

**ინფორმატიკისა და მართვის სისტემების ფაკულტეტი**

**პროექტი მონაცემთა ბაზებში**

**ჯგუფი:** 108740

**კურსი:** III

**სტუდენტები:** არჩილ ლომიძე, გიორგი ხითარიშვილი, თათია ოქროპირიძე, თორნიკე ტეტიაშვილი, ნინო შაინიძე

**თემა:** აფთიაქის კონცეპტუალური მოდელი

**ჩაბარების თარიღი:**

**ხელმძღვანელი:**

**თბილისი - 2020**

**სარჩევი**

**თავი 1. მონაცემთა ბაზის სტრუქტურის შემუშავება. მისი ობიექტების განსაზღვრა.**

**თავი 2. ცხრილების აღწერა და დაპროექტება, მიმდევრობების შექმნა**

**2.1 ცხრილის აღწერა**

**2.2 ცხრილების შექმნა, Constraint-ების დადგენა**

**2.3 ინდექსების განსაზღვრა**

**2.4 მიმდევრობების( Sequence) შექმნა**

**თავი 3. მონაცემების შეტანა ცხრილებში, მიმდევრობების გამოყენება**

***თავი 1. ამოცანის დასმა***

**ამოცანა: შეადგინეთ მონაცემთა ბაზის აფთიაქისათვის კონცეპტუ­ა­ლური მოდელი (მონაცემთა ბაზის სქემა). ბაზაში მოათავსეთ ინფორმაცია წამლების შესახებ და წამლების გაყიდვების შესახებ.**

**ა) შექმენით შესაბამისი ცხრილები და დაადგინეთ Constraint-ები. მოახდინეთ ცხრილებში საჭირო ველების ინდექსირება.**

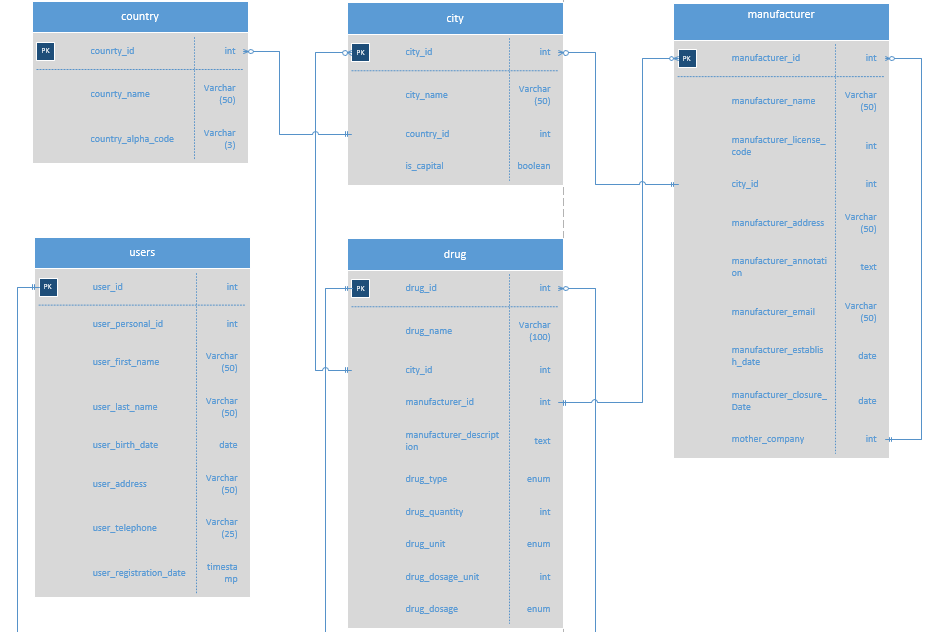
**ბ) შეავსეთ ცხრილები მონაცენებით. მონაცემების შეტანის დროს შექმენით და გამოიყენეთ Sequence-ები.**

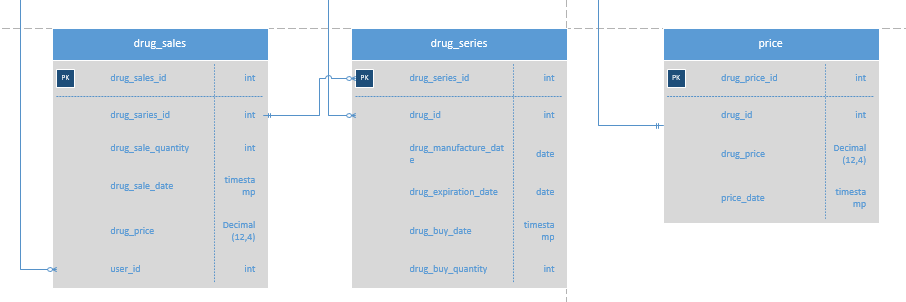
**გ) გამოიტანეთ მონაცემები წამლების შესახებ და დაალაგეთ წამლების დასახელების მიხედვით, წამლების დამზადების თარიღის მიხედვით. გამოიტანეთ ინფორმაცია წამლების გაყიდვების შესახებ და დაახარისხეთ წამლების გაყიდვის თარიღის მიხედვით.  იპოვეთ წამლების ფასებს შორის მაქსიმალური და მინიმალური და გამოიტანეთ ამ წამლების დასახელება და ფასი.**

**დ) შექმენით წარმოდგენა, რომელიც გამოიტანს ინფორმაციას წამლების შესახებ და წამლების დღიური გაყიდვების შესახებ ერთ ცხრილში.**

**ე) დაწერეთ  PL/SQL ბლოკი, რომელიც გამოიტანს წამლების დასახელებებს და შესაბამისი წამლების გაყიდვების შესახებ.**

***თავი 2. ცხრილების აღწერა და დაპროექტება***





●სურათზე მოცემულია მონაცემთა ბაზის კონცეპტუალური მოდელი

***2.1 ცხრილების აღწერა***

ბაზა შედგება 8 ცხრილისაგან:

◊ **Country** – ცხრილი შედგება ქვეყნების ჩამონათვალისგან:

**Country\_id**- ქვეყნის ნომერი (პირველადი გასაღები).

**Country\_name-** ქვეყნის დასახელება (უნიკალური).

**Country\_alpha\_code-** ქვეყნის ალფა კოდი.

◊ **City –** შეიცავს ინფორმაციას ქალაქების შესახებ.

**City\_id**- ქალაქის ნომერი (პირველადი გასაღები).

**City\_name-** ქალაქის დასახელება.

**Country\_id-** ქვეყნის ნომერი (მეორადი გასაღები).

**Is\_capital–** არის თუ არა დედაქალაქი.

◊ **Manufacture-** ცხრილი შეიცავს ინფორმაციას მწარმოებელი კომპანიების შესახებ:

**Manufacture\_id-** მწარმოებლის ნომერი (პირველადი გასაღები).

**Manufacture\_name-** მწარმოებლის სახელი.

**Manufacture\_licence\_code-** მწარმოებლის ლიცენზიის კოდი.

**City\_id-** ქალაქის ნომერი (მეორადი გასაღები).

**Manufacture\_address-** მწარმოებლის მისამართი.

**Manufacture\_annotation-** მწარმოებლის აღწერა.

**Manufacture\_email-** მწარმოებლის ელ-ფოსტა.

**Manufacture\_establish\_date-** მწარმოებლის დაარსების თარიღი.

**Manufacture\_closure\_date-** მწარმოებლის დახურვის თარიღი.

**Mother\_company-** მწარმოებლის მშობელი კომპანია.

◊ **Drug-** ცხრილი შეიცავს ინფორმაციას წამლების შესახებ:

**Drug\_id-** წამლების ნომერი (პირველადი გასაღები).

**Drug\_name-** წამლების სახელი

**City\_id-** ქალაქების ნომერი (მეორადი გასაღები).

**Manufacture\_id-** მწარმოებლის ნომერი (მეორადი გასაღები).

**Manufacture\_description-** მწარმოებლის მიერ გაცემული ანოტაცია.

**Drug\_type-** წამლების ტიპი.

**Drug\_quantity-** წამლების რაოდენობა.

**Drug\_unit-** წამლების ერთეული.

**Drug\_dosage\_unit-** წამლების დოზა კოლოფში.

**Drug\_dosage-** წამლების დოზა.

◊ **Users**- ცხრილი შედგება მომხმარებლების ინფორმაციისგან:

**User\_id-** მომხმარებლის ნომერი (პირველადი გასაღები).

**User\_personal\_id-** მომხმარებლის პირადი ნომერი.

**User\_first\_name-** მომხმარებლის სახელი.

**User\_last\_name-** მომხმარებლის გვარი.

**User\_birth\_date-** მომხმარებლის დაბადების თარიღი.

**User\_address-** მომხმარებლის მისამართი.

**User\_telephone-** მომხმარებლის ტელეფონი.

**User\_registration\_date-** მომხმარებლის რეგისტრაციის თარიღი.

◊ **Price-** ცხრილი შედგება წამლების ღირებულებებისგან:

**Drug\_price\_id**- ღირებულების ნომერი (პირველადი გასაღები).

**Drug\_id**- წამლის ნომერი (მეორადი გასაღები).

**Drug\_price**- წამლის ფასი.

**Price\_date**- ფასი კონკრეტული დღისთვის.

◊ **Drug\_series-** ცხრილი შედგება კომპანიის მიერ შეძენილი წამლების ინფორმაციებისგან:

**Drug\_series\_id-** შეძენილი წამლების ნომერი (პირველადი გასაღები).

**Drug\_id-** წამლების ნომერი(მეორადი გასაღები).

**Manufacture\_date -** წამლისწარმოების თარიღი.

**Drug\_expiration\_date-** წამლის მოქმედების ვადა.

**Drug\_buy\_date-** წამლის გაყიდვის თარიღი.

**Drug\_buy\_quantity-** გაყიდული წამლების რაოდენობა.

◊ **Drug\_sales-** ცხრილი შედგება წამლების გაყიდვის ინფორმაციისგან:

**Drug\_sales\_id-** გაყიდვის ნომერი (პირველადი გასაღები).

**Drug\_series\_id-** კომპანიის მიერ შეძენილი წამლის ნომერი (მეორადი გასაღები).

**Drug\_sales\_quantity-** გაყიდული წამლების რაოდენობა.

**Drug\_sale\_date-** გაყიდული წამლების თარიღი.

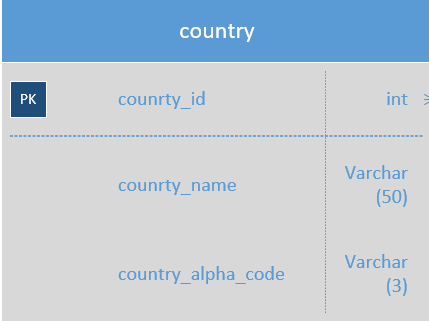
**Drug\_price-** წამლის ფასი.

**Sale\_user-** წამლის მყიდველი (მეორადი გასაღები).

***2.2 ცხრილების შექმნა, Constraint-ების დადგენა***

**•** ცხრილი **Country**: ქვემოთ მოცემულია ცხრილის სტრუქტურა და ბრძანება, რომლის

საშუალებითაც ხდება მისი შექმნა.



create table country (

country\_id int primary key not null,

country\_name varchar(50) unique not null,

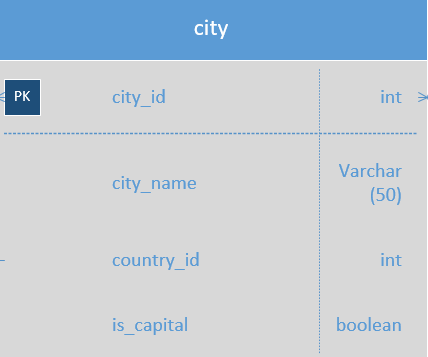
country\_alpha\_code varchar(3) unique not null

)engine=innodb;

ბრძანებაში **PRIMARY KEY(pk)** -ნიშნავს, რომ ველი ცხრილის პირველადი გასაღებია.

**VARCHAR2(50)-** მიუთითებს სერვერს, რომ ველი შეიცავს ტექსტურ ინფორმაციას და მისი მაქსიმალური სიგრძე 50 სიმბოლოა.

**●**ცხრილი **City:**

create tablecity (

city\_id int primary key not null,

city\_name varchar(50) not null,

country\_id int ,

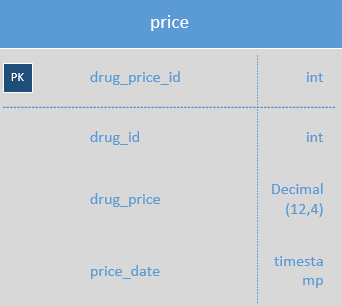
constraint foreign key(country\_id)

references country(country\_id),

is\_capital boolean

) engine=innodb;

●ცხრილი **price**:

create table price (

drug\_price\_id int primary key ,

drug\_id int ,

constraint foreign key (drug\_id)

references drug(drug\_id) ,

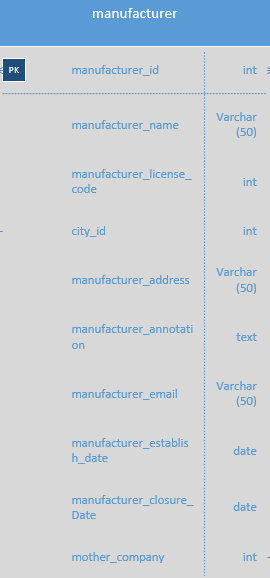
index drug\_price decimal(12,4) not null,

price\_date timestamp

)engine=innodb;

**Timestamp-ი** იღებს მნიშვნელობას ბრძანების გაშვების დროის მიხედვით.

**●**ცხრილი **manufacture: ●**ცხრილი **drug:**

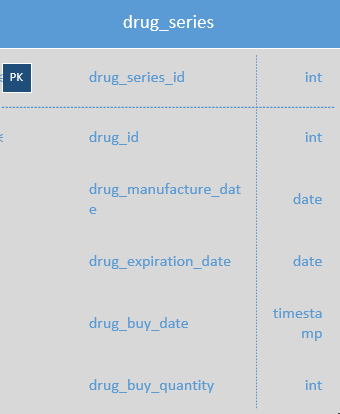
**Date-** ტიპი შეიცავს თარიღს.

**Enum-** ტიპი შეიცავს მნიშვნელობების სიისგან.

|  |  |
| --- | --- |
| create create table manufacturer (  manufacturer\_id int primary key ,  manufacturer\_name varchar(50) not null unique,  manufacturer\_license\_code int unique,  city\_id int ,  constraint foreign key (city\_id)  references city(city\_id),  manufacturer\_address varchar(50) not null,  manufacturer\_annotation text,  manufacturer\_email varchar(50) unique,  manufacturer\_establish\_date date not null ,  manufacturer\_closure\_date date,  mother\_company int ,  constraint foreign key(mother\_company)  references manufacturer(manufacturer\_id)  )engine=innodb; | create table drug(  drug\_id int primary key ,  drug\_name varchar(100) not null,  city\_id int,  constraint foreign key (city\_id)  references city(city\_id),  manufacturer\_id int ,  constraint foreign key (manufacturer\_id)  references manufacturer(manufacturer\_id),  manufacturer\_description text ,  drug\_type enum ('ointment','tablet', 'liquid'),  drug\_quantity int not null,  drug\_unit enum('pieces', 'grams') ,  drug\_dosage\_unit int ,  drug\_dosage enum ('grams', 'milligrams')  )engine=innodb; |
| **●**ცხრილი **drug\_sales:** | **●ცხრილი users:** |

|  |  |
| --- | --- |
| create table drug\_sales (  drug\_sales\_id int primary key,  drug\_series\_id int ,  constraint foreign key (drug\_series\_id)  references drug\_series(drug\_series\_id),  drug\_sale\_quantity int not null,  drug\_sale\_date timestamp,  drug\_price decimal(12,4),  sale\_user int ,  constraint foreign key (sale\_user)  references users(user\_id)  )engine=innodb; | create table users(  user\_id int primary key,  user\_personal\_id int not null unique,  user\_first\_name varchar(50) not null,  user\_last\_name varchar(50) not null,  user\_birth\_date date ,  user\_address varchar(50) ,  user\_telephone varchar(25) not null,  user\_registration\_date timestamp  )engine=innodb; |

**●ცხრილი drug\_series:**



|  |
| --- |
| create table drug\_series(  drug\_series\_id int primary key ,  drug\_id int ,  constraint foreign key (drug\_id)  references drug(drug\_id) ,  drug\_manufacture\_date date not null,  drug\_expiration\_date date not null,  drug\_buy\_date timestamp,  drug\_buy\_quantity int not null  )engine=innodb; |

**2.3 ინდექსების განსაზღვრა**

***რა არის ინდექსი ?***

ინდექსი არის მონაცემთა ბაზის ობიექტი, რომლის დანიშნულებაც არის ბაზიდან მონაცემების ამორჩევის დაჩქარება. ხოლო, თუკი შესაბამის ველზე მითითებული იქნება ინდექსი, სერვერი მონაცემებს შეინახავს ისეთ ფორმატში, რომელიც შემდგომში საშუალებას მისცემს ყველა ჩანაწერის გადარჩევის ნაცვლად გამოიყენოს განსაკუთრებული ალგორითმი, რათა შეასრულოს მინიმალური რაოდენობის ოპერაცია(შედარება). თუ მითითებული იქნება UNIQUE (უნიკალური) ინდექსი, სერვერი მნიშვნელობების პირველი დამთხვევისთანავე დააბრუნებს შედეგს და აღარ განაგრძნობს დანარჩენი ჩანაწერების შემოწმებას, ვინაიდან UNIQUE ველის მნიშვნელობა ცხრილში არ მეორდება.

**2.4 მიმდევრობების( Sequence) შექმნა**

**რა არის მიმდევრობა ?**

მიმდევრობა(**sequence**) - წარმოადგენს ობიექტს, რომელიც გამოიყენება რიცხვების გენერაციისთვის, მითითებული წესების თანახმად. მიმდევრობა შეიძლება გამოსადეგი იყოს პირველადი გასაღების ან სხვა უნიკალური რიცხვის გენერაციისთვის. | მიმდევრობას აქვს ორი ფსევდოსვეტი **curreal** და **nextval**, რომლებიც აბრუნებენ მიმდევრობის მიმდინარე და მომდევნო მნიშვნელობას შესაბამისად, **nextval** მნიშვნელობის გამოყენებისას, მიმდევრობა გადადის შემდეგ მნიშვნელობაზე.

მიმდევრობას გააჩნია სახელი და რამდენიმე პარამეტრი, მათ შორის:

**START WITH** - მიმდევრობის საწყისი მნიშვნელობა

**INCREMENT BY** - მიმდევრობის ბიჯი, განსაზღვრავს, თუ რამდენით უნდა იცვლებოდეს მიმდევრობის ყოველი მომდევნო მნიშვნელობა.

**MINVALUE, MAXVALUE** - მიმდევრობის მინ. და მაქს. მნიშვნელობები,

**CYCLE, NOCYCLE** - უნდა დაიწყოს თუ არა თავიდან ათვლა, მინ. ან მაქს. მნიშვნელობის მიღწევისას

**CACHE, NOCACHE** - უნდა მოხდეს თუ არა მიმდევრობის მომდევნო რამდენიმე წევრის ქეშირება(წინასწარ გამოთვლა და დამახსოვრება).

ქვემოთ მითითებულია ბრძანებები მიმდევრობების შესაქმნელად.

**ქვეყნების მიმდევრობა:**

create sequences country\_c

increment by 1

start with 0

minvalue 0

maxvalue 999

nocycle

nocache;

ბრძანება ქმნის მიმდევრობას სახელად Quch\_seq, რომლის საწყისი მნიშვნელობა და ბიჯი 1 ს ტოლია, რაც იმას ნიშნავს, რომ რიცხვის ყოველი გენერირებისას მნიშვნელობა გაიზრდება 1 ერთეულით.

**ქალაქების მიმდევრობა:**

create sequences city\_cit

increment by 1

start with 0

minvalue 0

maxvalue 999

nocycle

nocache;

**მომხმარებლების მიმდევრობა:**

create sequences user\_u

increment by 1

start with 0

minvalue 0

maxvalue 999

nocycle

nocache;

**თავი 3. მონაცემების შეტანა ცხრილებში, მიმდევრობების გამოყენება**

მონაცემების შეტანა Country ცხრილში:

|  |
| --- |
| INSERT INTO country (country\_id, country\_name, country\_alpha\_code)  VALUES ('1', 'Georgia','GE'), ('2', 'Japan', 'JP'); |

მონაცემების შეტანა City ცხრილში:

|  |
| --- |
| INSERT into city(city\_id, city\_name, country\_id, is\_capital)  VALUES ('1', 'Tbilisi', '1', '1'), ('2', 'Tokyo', '2', '1'); |

მონაცემების შეტანა Manufacture ცხრილში:

|  |
| --- |
| INSERT into manufacturer  VALUES ('1','BTG', '657','1','Pekini #5', 'am wamalma sheidzleba gamoiwvios guldzmarva', 'marting@gmail.com', '1986-12-02', '2000-01-01','1'),  ('2','Ego','453','3','kofuku #12','ukuchvnebebi: halucinaciebi, gulisreva da udziloba', 'egodisco@gmail.com', '2002-04-30',null,'2'); |

მონაცემების შეტანა Drug ცხრილში:

|  |
| --- |
| INSERT into drug(drug\_id, drug\_name, city\_id, manufacturer\_id, manufacturer\_description, drug\_type, drug\_quantity, drug\_unit, drug\_dosage\_unit,drug\_dosage)  VALUES  ('1','ketotifeni','3','2','es kompania gamoircheva siswrafit da xarisxiani produqtit','tablet','50','pieces','2000','grams'), |

მონაცემების შეტანა Users ცხრილში:

|  |
| --- |
| INSERT into users(user\_id, user\_personal\_id, user\_first\_name, user\_last\_name, user\_birth\_date, user\_address, user\_telephone, user\_registration\_date)  VALUES  ('1',null,'Tatia', 'Oqropiridze','1999-11-01 00:00:00','pekinis 21','555566302',null),  ('2',null,'Giorgi', 'Khitarishvili','1998-12-28 00:00:00','vaja pshavelas 7','591953681',null), |

მონაცემების შეტანა Price ცხრილში:

|  |
| --- |
| INSERT into price(drug\_price\_id, drug\_id, drug\_price, price\_date)  VALUES  ('1','6','15.7',null),  ('2','4','12.89',null),  ('3','2','6.12',null), |

მონაცემების შეტანა Drug\_Sales ცხრილში:

|  |
| --- |
| INSERT into drug\_sales(drug\_sales\_id, drug\_series\_id, drug\_sale\_quantity, drug\_sale\_date, drug\_buy\_date, drug\_prise, sale\_user)  VALUES  ('1','4','23','2013-09-11','2010-07-04','15.7','4'),  ('2','1','30','2004-07-12','2003-09-08','12.89','3'),  ('3','2','20','2019-01-03','2018-05-05','6.12','20'), |

მონაცემების შეტანა Drug\_Series ცხრილში:

|  |
| --- |
| INSERT into drug\_series(drug\_series\_id, drug\_id, drug\_manufacture\_Date, drug\_expiration\_date, drug\_buy\_date, drug\_buy\_quantity)  VALUES  ('1','5','2001-11-02','2006-12-20',null,'3000'),  ('2','3','2000-11-11','2003-12-31',null,'5000') |

Pl/sql ის ბლოკი (ჩონჩხი)

DECLARE

BEGIN

SELECT \*

FROM drug d

LEFT JOIN drug\_sales ds

ON ds.drug\_sales\_id=d.drug\_id;

dbms\_output.put\_line(drug\_name);

dbms\_output.put\_line(drug\_sale);

END;