Маратулы Темирболат

Лабораторная работа № 2

1. Создать НОРМАЛИЗОВАННУЮ базу данных с первичными ключами, для системы скорой помощи:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Работник** | **Бриагада** | **Дата и время вызова** | **Адрес** | **ФИО пациента, год рождения** | **Причина вызова** | Предварительный диагноз | Результат |
| Иванова Анна | 1 | 12.07.2017 13:00 | Монструозная, 25 кв.1 | Гдлян, О.И, 1957 г.р. | Боль в области сердца | ИБС | Оказана помощь на месте |
| Сидоров Сергей | 2 | 13.08.2017 14:35 | Спасательная, 61 | Ашимова Н.Н., 1990 г.р | Боль в правой подвздошной области | Аппендицит | Доставлен в 1 гор. Больницу |
| Ахметова Динара | 1 | 14.08.2017 13:00 | Кантарбекова, 29/4, кв.1 | Курсова К.Д., 1985 г.р. | Ножевое ранение, кровопотеря | Ножевое ранение, гемодинамический шок | Оказана первая помощь, Доставлен в ОКБ |
| Магзумов Асет | 3 | 15.08.2017 10:20 | Олжасова 21-б, кв. 235 | Атаева У.К, 1975 г.р. | Потеря сознания |  | Оказана помощь на месте, доставлен в 1 гор.больницу |

Diagram

Description automatically generated

1. Для нормализованных таблиц создать внешние ключи, ограничения должны содержать разные значения ON DELETE;

CREATE TABLE brigade  
(  
 id serial PRIMARY KEY  
);  
  
CREATE TABLE employee  
(  
 id serial PRIMARY KEY,  
 name varchar(20) NOT NULL,  
 surname varchar(20) NOT NULL,  
 brigade\_id integer NOT NULL,  
 CONSTRAINT fk\_brigade\_id FOREIGN KEY (brigade\_id) REFERENCES brigade (id) ON DELETE NO ACTION  
);  
  
CREATE TABLE patient  
(  
 id serial PRIMARY KEY,  
 first\_name varchar(40) NOT NULL,  
 last\_name varchar(40) NOT NULL,  
 middle\_name varchar(40),  
 date\_birth date CHECK (date\_birth >= '1900.01.01')  
);  
  
CREATE TABLE diagnosis  
(  
 id serial PRIMARY KEY,  
 name varchar(50) UNIQUE NOT NULL  
);  
  
CREATE TABLE call  
(  
 id serial PRIMARY KEY,  
 employee\_id integer NOT NULL,  
 patient\_id integer NOT NULL,  
 street varchar(30) NOT NULL,  
 house\_number varchar(5) NOT NULL,  
 flat\_number integer CHECK ( flat\_number > 0 ),  
 date\_time timestamp NOT NULL,  
 CONSTRAINT fk\_emploee\_id FOREIGN KEY (employee\_id) REFERENCES employee (id) ON DELETE RESTRICT,  
 CONSTRAINT fk\_patient\_id FOREIGN KEY (patient\_id) REFERENCES patient (id) ON DELETE SET NULL  
);  
  
CREATE TABLE reason\_diagnosis(  
 id serial PRIMARY KEY,  
 call\_id integer NOT NULL,  
 reason varchar(50) NOT NULL,  
 provisional\_diagnosis\_id integer NOT NULL,  
 CONSTRAINT fk\_call\_id FOREIGN KEY (call\_id) REFERENCES call(id) ON DELETE CASCADE,  
 CONSTRAINT fk\_diagnosis\_id FOREIGN KEY (provisional\_diagnosis\_id) REFERENCES diagnosis(id) ON DELETE RESTRICT  
);  
  
CREATE TABLE call\_result  
(  
 id serial PRIMARY KEY,  
 call\_id integer NOT NULL,  
 call\_result varchar(50) NOT NULL,  
 CONSTRAINT fk\_call\_id FOREIGN KEY (call\_id) REFERENCES call (id) ON DELETE CASCADE  
);

1. Внести тестовые данные в каждую таблицу;

INSERT INTO brigade  
VALUES (1),  
 (2),  
 (3),  
 (4),  
 (5);  
  
INSERT INTO employee(name, surname, brigade\_id)  
VALUES ('Анна', 'Иванова', 1),  
 ('Сергей', 'Сидоров', 2),  
 ('Динара', 'Ахметова', 1),  
 ('Асет', 'Магзумов', 3);  
  
INSERT INTO patient(first\_name, last\_name, middle\_name, date\_birth)  
VALUES ('Олег', 'Гдлян', 'Игоревич', '1957.12.10'),  
 ('Николай', 'Ашимова', 'Игоревна', '1995.01.31'),  
 ('Карина', 'Курсова', 'Домодедовна', '1952.10.23'),  
 ('Улсая', 'Атаева', 'Кадырбековна', '1999.05.05');  
  
INSERT INTO diagnosis(name)  
VALUES ('ИБС'),  
 ('Аппендицит'),  
 ('Ножевое ранение'),  
 ('Гемодинамический шок');  
  
INSERT INTO diagnosis(name) VALUES ('');  
  
INSERT INTO call(employee\_id, patient\_id, street, house\_number, flat\_number, date\_time)  
VALUES (1, 1, 'Монструозная', '25', 1, '12.07.2017 13:00'),  
 (2, 2, 'Спасательная', '61', NULL, '13.08.2017 14:35'),  
 (3, 3, 'Кантарбекова', '29/4', '1', '14.08.2017 13:00'),  
 (4, 4, 'Олжасова', '21-6', 235, '15.08.2017 10:20');  
  
INSERT INTO reason\_diagnosis(call\_id, reason, provisional\_diagnosis\_id) VALUES (1,'Боль в области сердца',1),  
 (2,'Боль в правой подвздошной области',2),  
 (3,'Ножевое ранение',3),  
 (3,'Кровопотеря',4),  
 (4,'Потеря сознания',5);  
  
INSERT INTO call\_result(call\_id, call\_result)  
VALUES (1, 'Оказана помощь на месте'),  
 (2, 'Доставлен в 1 гор. Больницу'),  
 (3, 'Оказана первая помощь'),  
 (3, 'Доставлен в ОКБ'),  
 (4, 'Оказана помощь на месте'),  
 (4, 'Доставлен в 1 гор.больницу');

1. Создать представление содержащее список бригад с количеством вызовов за сентябрь 2017 года;

CREATE VIEW brigade\_call\_september AS  
SELECT b.id AS brigade\_number, *COUNT*(c.id) AS call\_number  
FROM call c  
 JOIN employee e  
 ON c.employee\_id = e.id  
 JOIN brigade b  
 ON b.id = e.brigade\_id  
WHERE c.date\_time BETWEEN '2017.09.01' AND '2017.09.30'  
GROUP BY b.id;

Graphical user interface, application

Description automatically generated

1. Создать представление, содержащее больных с датой и причиной вызова;
2. CREATE VIEW patient\_call\_reason AS  
   SELECT *CONCAT*(p.first\_name, ' ', p.last\_name, ' ', p.middle\_name) AS fio\_patient, c.date\_time, r\_d.reason  
   FROM call c  
    JOIN patient p  
    ON c.patient\_id = p.id  
    JOIN reason\_diagnosis r\_d  
    ON c.id = r\_d.call\_id;

A picture containing graphical user interface

Description automatically generated

1. Создать представление, содержащее список в точности как в таблице п.1.

CREATE VIEW full\_info AS  
SELECT *CONCAT*(e.name,' ',e.surname) AS employee\_fio,e.brigade\_id,date\_time,*CONCAT*(c.street,',д.',c.house\_number,',',c.flat\_number) AS adress,  
 *CONCAT*(p.first\_name,' ',p.last\_name,' ',p.middle\_name,',',date\_birth) AS patient\_fio\_birthday,r\_d.reason,diag.name,call\_result  
FROM call c  
 JOIN call\_result cr

ON c.id = cr.call\_id  
 JOIN patient p

ON c.patient\_id = p.id  
 JOIN employee e

ON e.id = c.employee\_id  
 JOIN reason\_diagnosis r\_d

ON r\_d.call\_id = c.id  
 JOIN diagnosis diag

ON diag.id = r\_d.provisional\_diagnosis\_id;

Diagram

Description automatically generated

1. Создать представление, включающее: улиц и количество вызовов по каждой улице;

CREATE VIEW street\_info AS  
SELECT street, *COUNT*(id) AS calls\_number  
FROM call  
GROUP BY street;

Table

Description automatically generated