

SUG'URTA KOMPANIYASI MIJOZLARINI BOSHQARISH TIZIMI: SQL ASOSIDA AMALIY YONDASHUV

*Farg'ona davlat universiteti
o'qituvchisi Tojimatov Israil Nurmamatovich
FarDU Fizika-matematika fakulteti 23.11-guruh
talabasi Xamidov Hojiakbar Abdurasilovich
hojiakbarxamidov1@gmail.com*

Annotatsiya

Ushbu maqolada sug'urta kompaniyasining mijozlarini boshqarish tizimi uchun ma'lumotlar bazasi loyihasi ishlab chiqildi. Maqolada SQL asosida jadvallar yaratish, ma'lumotlarni kiritish va tahlil qilish bo'yicha amaliy misollar keltirilgan. Taklif etilgan yechim sug'urta kompaniyalarining mijozlar, polislar, to'lovlar va da'volarni samarali boshqarishida qo'l keladi.

Kalit so'zlar: Sug'urta kompaniyasi, mijozlar bazasi, SQL, ma'lumotlar bazasi, sug'urta polisi, to'lovlar, da'volar, axborot tizimi, tahliliy so'rovlar

Аннотация

В данной статье разработан проект базы данных для системы управления клиентами страховой компании. Представлены практические примеры создания таблиц, ввода и анализа данных с использованием SQL. Предложенное решение позволяет эффективно управлять клиентами, полисами, платежами и страховыми случаями.

Ключевые слова: Страховая компания, база клиентов, SQL, база данных, страховой полис, платежи, страховые случаи, информационная система, аналитические запросы

Annotation

This article presents a database design for an insurance company's client management system. It includes practical examples of creating tables, inserting data, and analyzing it using SQL. The proposed solution helps effectively manage clients, insurance policies, payments, and claims.

Keywords: Insurance company, client database, SQL, database, insurance policy, payments, claims, information system, analytical queries

Kirish

Bugungi raqamli davrda ko'plab tashkilotlar o'z faoliyatini optimallashtirish va ma'lumotlar bilan samarali ishlash uchun axborot texnologiyalaridan foydalanmoqda. Ayniqsa, sug'urta kompaniyalari uchun mijozlar haqidagi to'liq, aniq va tizimli ma'lumotlar bazasining mavjudligi muhim hisoblanadi. Har bir mijozning sug'urta

polisi, to'lovlar tarixi, da'volari (claim) va boshqa ma'lumotlarini bir joyda saqlash – tezkor xizmat ko'rsatish, tahlil qilish hamda strategik qarorlar qabul qilish imkonini beradi.

Ushbu maqolada sug'urta kompaniyasi uchun berilganlar bazasini loyihalash, SQL tili yordamida jadvallar yaratish, ma'lumot kiritish va ularni tahlil qilish misollar asosida ko'rib chiqiladi. Bu tizim kichik yoki o'rta hajmdagi kompaniyalar uchun foydali yechim bo'lishi mumkin.

Berilganlar bazasi modeli

Mijozlar bilan ishlovchi sug'urta kompaniyasi uchun quyidagi asosiy jadvallarni yaratish maqsadga muvofiq:

1. Mijozlar (Clients) – mijozlar haqida asosiy ma'lumotlar
 2. Sug'urta_polislari (Policies) – mijozga tegishli sug'urta polis(sertifikat)lari
 3. To'lovlar (Payments) – sug'urta uchun to'lovlar tarixi
 4. Da'volar (Claims) – mijoz tomonidan sug'urta da'volari
- SQL kodlari bilan amaliy misollar

1. Jadval yaratish

-- Mijozlar jadvali

```
CREATE TABLE Clients (
  client_id INT PRIMARY KEY,
  full_name VARCHAR(100),
  passport_number VARCHAR(20),
  phone_number VARCHAR(20),
  address TEXT
);
```

-- Sug'urta polislari jadvali

```
CREATE TABLE Policies (
  policy_id INT PRIMARY KEY,
  client_id INT,
  policy_type VARCHAR(50),
  issue_date DATE,
  expiry_date DATE,
  premium_amount DECIMAL(10, 2),
  FOREIGN KEY (client_id) REFERENCES Clients(client_id)
);
```

-- To'lovlar jadvali

```
CREATE TABLE Payments (  
    payment_id INT PRIMARY KEY,  
    client_id INT,  
    payment_date DATE,  
    amount DECIMAL(10, 2),  
    FOREIGN KEY (client_id) REFERENCES Clients(client_id)  
);
```

-- *Da'volar jadvali*

```
CREATE TABLE Claims (  
    claim_id INT PRIMARY KEY,  
    client_id INT,  
    claim_date DATE,  
    claim_amount DECIMAL(10, 2),  
    claim_status VARCHAR(20),  
    FOREIGN KEY (client_id) REFERENCES Clients(client_id)  
);
```

2. Ma'lumot kiritish (INSERT INTO)

```
INSERT INTO Clients VALUES (1, 'Diyor Karimov', 'AA1234567',  
    '+998901234567', 'Toshkent shahri');  
INSERT INTO Policies VALUES (101, 1, 'Avtomobil', '2024-01-01',  
    '2025-01-01', 1500000.00);  
INSERT INTO Payments VALUES (201, 1, '2024-01-02',  
    1500000.00);  
INSERT INTO Claims VALUES (301, 1, '2024-06-01', 300000.00,  
    'Ko'rib chiqilmoqda');
```

3. Oddiy so'rovlar (SELECT)

-- *Barcha mijozlar ro'yxatini olish*

```
SELECT * FROM Clients;
```

-- *Ma'lum polis turi bo'yicha mijozlar*

```
SELECT c.full_name, p.policy_type  
FROM Clients c  
JOIN Policies p ON c.client_id = p.client_id  
WHERE p.policy_type = 'Avtomobil';
```

```
-- Oxirgi 6 oy ichida da'vo bildirgan mijozlar
SELECT c.full_name, cl.claim_date, cl.claim_amount
FROM Clients c
JOIN Claims cl ON c.client_id = cl.client_id
WHERE cl.claim_date >= DATE_SUB(CURDATE(), INTERVAL 6
MONTH);
```

4. Tahliliy SQL so'rovlar

```
-- Har bir mijoz qancha sug'urta mukofoti to'laganini chiqarish
SELECT c.full_name, SUM(p.amount) AS total_payment
FROM Clients c
JOIN Payments p ON c.client_id = p.client_id
GROUP BY c.full_name
ORDER BY total_payment DESC;
```

```
-- Da'vo holatlari bo'yicha statistika
SELECT claim_status, COUNT(*) AS total_claims
FROM Claims
GROUP BY claim_status;
```

```
-- Har bir polis turiga nechta mijoz ega
SELECT policy_type, COUNT(*) AS total_clients
FROM Policies
GROUP BY policy_type;
```

Xulosa

Ushbu maqolada sug'urta kompaniyasining mijozlarini boshqarish tizimi uchun berilganlar bazasini yaratish jarayoni yoritildi. SQL tili yordamida asosiy jadvallar yaratildi, ularga ma'lumotlar kiritildi va foydali so'rovlar amalga oshirildi. Ushbu tizim orqali kompaniya mijozlar bilan bog'liq barcha asosiy ma'lumotlarni markazlashtirilgan tarzda boshqarishi, moliyaviy hisobotlar tayyorlashi va xizmat sifatini oshirishi mumkin.

Tizim kelgusida kengaytirilishi mumkin: foydalanuvchi rollari, sug'urta voqealarining tasdiqlanish jarayonlari, xavfsizlik siyosatlari kabi qo'shimchalar qo'shilishi mumkin.

Foydalanilgan adabiyotlar

1. Elmas Raxmatov – “Ma’lumotlar bazasini boshqarish tizimlari”, Toshkent, 2022.
2. MySQL rasmiy hujjatlari: <https://dev.mysql.com/doc/>
3. W3Schools SQL Tutorial: <https://www.w3schools.com/sql/>
4. PostgreSQL Documentation: <https://www.postgresql.org/docs/>