INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCHERS
ISSN: 3030-332X Impact factor: 9,493 Volume 17, issue 1, Aprel 2026
https://wordlyknowledge.uz/index.php/IJSR worldly knowledge
Index: google scholar, research gate, research bib, zenodo, open aire.
https://scholar.google.com/scholar?hl=ru&as_sdt=0%2C5&q=wosjournals.com&btnG
https://www.researchgate.net/profile/Worldly-Knowledge
https://journalseeker.researchbib.com/view/issn/3030-332X

ANDROID STUDIO MUHITIDA SQLITE MA'LUMOTLAR BAZASIDAN FOYDALANISH

Omonova Zarnigor Shukrullo qizi
Andijon davlat texnika instituti talabasi
https://doi.org/10.5281/zenodo.19946550

Annotatsiya: Zamonaviy mobil ilovalar ko'pincha tarmoq ulanishisiz ham ma'lumotlarni saqlash, qayta ishlash va foydalanuvchiga taqdim etish imkoniyatini talab qiladi. Bunday holatlarda lokal ma'lumotlar bazasi (Local Database) yechimi sifatida eng keng qo'llaniladigan texnologiyalardan biri SQLite hisoblanadi. SQLite – bu nol konfiguratsion, server talab qilmaydigan, fayl asosida ishlaydigan yengil relatsion ma'lumotlar bazasi bo'lib, Android operatsion tizimiga standart kutubxona sifatida integratsiya qilingan. Ushbu maqolaning asosiy maqsadi – Android Studio muhitida SQLite yordamida ma'lumotlar bazasini yaratish, CRUD (Create, Read, Update, Delete) operatsiyalarini amalga oshirish va ishlab chiqish jarayonidagi amaliy muammolarni hal qilish usullarini tizimli ravishda ko'rsatishdir. Tadqiqot obyekti sifatida Android ilovalarida lokal ma'lumotlarni boshqarish arxitekturasi, predmeti esa SQLite yordamida samarali va xavfsiz ma'lumotlar oqimini tashkil etish metodlari belgilangan.
Kalit so'zlar: Android Studio, SQLite ma'lumotlar bazasi, lokal saqlash, CRUD operatsiyalari, SQLiteOpenHelper, Kotlin, MVVM arxitektura, asinxron dasturlash.

Abstract: Modern mobile applications often require the ability to store, process, and deliver data to users even in the absence of a network connection. In such scenarios, SQLite stands out as one of the most widely adopted technologies for local database solutions. SQLite is a zero-configuration, serverless, file-based, lightweight relational database that is integrated into the Android operating system as a standard library. The primary objective of this paper is to systematically demonstrate the process of creating a database using SQLite within the Android Studio environment, implementing CRUD (Create, Read, Update, Delete) operations, and addressing practical challenges that arise during the development process. The research object is defined as the architecture of local data management in Android applications, while the research subject focuses on methodologies for establishing an efficient and secure data flow using SQLite.
Keywords: Android Studio, SQLite database, local storage, CRUD operations, SQLiteOpenHelper, Kotlin, MVVM architecture, asynchronous programming.

Аннотация: Современные мобильные приложения часто требуют возможности хранения, обработки и предоставления данных пользователю даже при отсутствии сетевого подключения. В таких ситуациях в качестве решения для локальной базы данных (Local Database) одной из наиболее широко применяемых технологий является SQLite. SQLite — это легковесная реляционная база данных, работающая на основе файлов, не требующая настройки конфигурации и сервера, которая интегрирована в операционную систему Android в виде стандартной библиотеки. Основной целью данной статьи является систематическое демонстрация процесса создания базы данных с помощью SQLite в среде Android Studio, реализации CRUD-операций (Create, Read, Update, Delete) и методов решения практических задач, возникающих в процессе разработки. В качестве объекта исследования определена архитектура управления локальными данными в Android-приложениях, а предметом исследования выступают

методы организации эффективного и безопасного потока данных с использованием SQLite.

Ключевые слова: Android Studio, база данных SQLite, локальное хранение, CRUD-операции, SQLiteOpenHelper, Kotlin, MVVM-архитектура, асинхронное программирование.

Bugungi kunda mobil texnologiyalar jadal rivojlanib, Android operatsion tizimi asosida yaratilgan ilovalar kundalik hayotimizning ajralmas qismiga aylandi. Mobil ilovalarda foydalanuvchi ma’lumotlarini saqlash muhim vazifalardan biri hisoblanadi. Ushbu ehtiyojni qondirish uchun Android platformasida bir nechta ma’lumot saqlash usullari mavjud bo‘lib, ulardan eng keng tarqalgani SQLite ma’lumotlar bazasidir.

SQLite — bu server talab qilmaydigan, yengil va tezkor ma’lumotlar bazasi bo‘lib, Android tizimiga standart tarzda integratsiyalashgan. Ushbu maqolaning asosiy maqsadi Android Studio muhitida SQLite’dan foydalanish jarayonini ilmiy va amaliy jihatdan tahlil qilishdan iborat.

Metodlar (methods)

Tadqiqot Android Studio (Iguana/Hedgehog versiyalari), Kotlin dasturlash tili va Android SDK 34 platformasi asosida olib borildi. Loyiha arxitekturasida MVVM (Model-View-ViewModel) naqshidan foydalanildi. SQLite bilan ishlash uchun SQLiteOpenHelper klassidan meros olingan AppDatabase komponenti yaratildi. Unda quyidagi metodlar amaliyotga tatbiq etildi:

- onCreate(SQLiteDatabase db): jadval tuzilmasini CREATE TABLE SQL buyrug‘i orqali yaratish;
- onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion): versiya oshganda jadvalni o‘zgartirish yoki qayta qurish;
- insertData(), getAllData(), updateData(), deleteData(): CRUD operatsiyalari uchun alohida metodlar.

Barcha ma’lumotlar bazasi operatsiyalari asosiy UI threadni bloklamaganligi uchun kotlinx.coroutines kutubxonasi yordamida Dispatchers.IO kontekstida asinxron bajarildi. UI qismida RecyclerView, EditText va MaterialButton komponentlari, ma’lumotlar oqimi uchun esa StateFlow qo‘llanildi. Sinov jarayoni Pixel 6 emulyatori (API 33) va fizik Samsung A14 qurilmasida olib borildi. Ma’lumotlar tuzilmasi va yozuvlar to‘g‘riligi Android Studio Database Inspector vositasi, ish vaqti va xotira iste’moli esa Logcat hamda Android Profiler orqali kuzatildi. Har bir metod try-catch bloklari va SQLiteDatabase.CONFLICT_REPLACE strategiyasi bilSQLite Android ekotizimida lokal ma’lumotlarni saqlashning eng ishonchli va tarmoqqa bog‘liq bo‘lmagan yechimlaridan biri hisoblanadi. Uning afzalliklari: tashqi server yoki kutubxona talab qilmaslik, SQL standartlariga to‘liq moslik, kichik binari o‘lcham va yuqori tezlik. Biroq, to‘g‘ridan-to‘g‘ri SQLiteOpenHelper bilan ishlashda quyidagi kamchiliklar kuzatiladi: qo‘lda yozilgan SQL so‘rovlarida sintaksis xatolarining compile-time da aniqlanmasligi, migratsiya jarayonining qo‘lda boshqarilishi, va xavfsizlik (SQL injection) xavfi mavjud.

Zamonaviy Android ishlab chiqishda Google rasmiy ravishda Room Persistence Library ni tavsiya qiladi. Room SQLite ustiga qurilgan abstraktsiya bo‘lib, @Dao, @Entity, @Database annotatsiyalari orqali tip-xavfsiz so‘rovlarni, LiveData/Flow integratsiyasini va avtomatik migratsiyani qo‘llab-quvvatlaydi. Ushbu tadqiqot natijalari shuni ko‘rsatadiki, oddiy yoki prototip loyihalar uchun to‘g‘ridan-to‘g‘ri SQLite yetarli, ammo tijorat yoki murakkab

arxitekturali ilovalarda Room ga o‘tish kod sifatini, test qoplanishini va qo‘llab-quvvatlash qulayligini sezilarli oshiradi. Kelgusi tadqiqotlarda Room + Coroutines + Flow integratsiyasi, MockK yordamida unit testing va AutoMigration strategiyalarini chuqurroq o‘rganish rejalashtirilgan.an xatoliklarni boshqarish mexanizmlari bilan mustahkamlandi.

Xulosa

Android Studio muhitida SQLite ma’lumotlar bazasidan foydalanish lokal ma’lumotlarni boshqarishning samarali, ishonchli va tarmoqqa bog‘liq bo‘lmagan usuli hisoblanadi. To‘g‘ri arxitektura (MVVM), asinxron ishlov berish (Coroutines), xatoliklarni boshqarish va zamonaviy diagnostika vositalaridan foydalanish orqali barqaror va foydalanuvchiga qulay ilovalar yaratish mumkin. Dasturchilar loyiha murakkabligiga qarab to‘g‘ridan-to‘g‘ri SQLite yoki Room kutubxonasini tanlashlari, har doim background threadlarda DB operatsiyalarini bajarishlari va migratsiya jarayonlarini oldindan rejalashtirishlari tavsiya etiladi.

Ilmiy rahbar: Dehqonov Tavakkaljon

Foydalanilgan adabiyotlar

1. Android Developers Documentation. Save data using SQLite. Android Platform Documentation, 2025.
2. Android Developers. Room Persistence Library – Jetpack. Android Developers, 2024.
3. SQLite Official Documentation. SQLite Version 3: Core Features and SQL Syntax. SQLite.org, 2025.
4. Android Developers. Persist data with Room – Codelab. Android Codelabs, 2024.
5. Kotlinlang Documentation. Kotlin Coroutines Guide: Asynchronous Programming. JetBrains, 2025.
6. Android Developers Documentation. Save data using SQLite. Android Platform Documentation, 2025.
7. Android Developers. Room Persistence Library – Jetpack. Android Developers, 2024.
8. SQLite Official Documentation. SQLite Version 3: Core Features and SQL Syntax. SQLite.org, 2025.
9. Android Developers. Persist data with Room – Codelab. Android Codelabs, 2024.
10. Kotlinlang Documentation. Kotlin Coroutines Guide: Asynchronous Programming. JetBrains, 2025.
11. Android Developers Documentation. Save data using SQLite. Android Platform Documentation, 2025.
12. Android Developers. Room Persistence Library – Jetpack. Android Developers, 2024.
13. SQLite Official Documentation. SQLite Version 3: Core Features and SQL Syntax. SQLite.org, 2025.
14. Android Developers. Persist data with Room – Codelab. Android Codelabs, 2024.
15. Kotlinlang Documentation. Kotlin Coroutines Guide: Asynchronous Programming. JetBrains, 2025.
16. Android Developers Documentation. Save data using SQLite. Android Platform Documentation, 2025.
17. Android Developers. Room Persistence Library – Jetpack. Android Developers, 2024.
18. SQLite Official Documentation. SQLite Version 3: Core Features and SQL Syntax. SQLite.org, 2025.
19. Android Developers. Persist data with Room – Codelab. Android Codelabs, 2024.