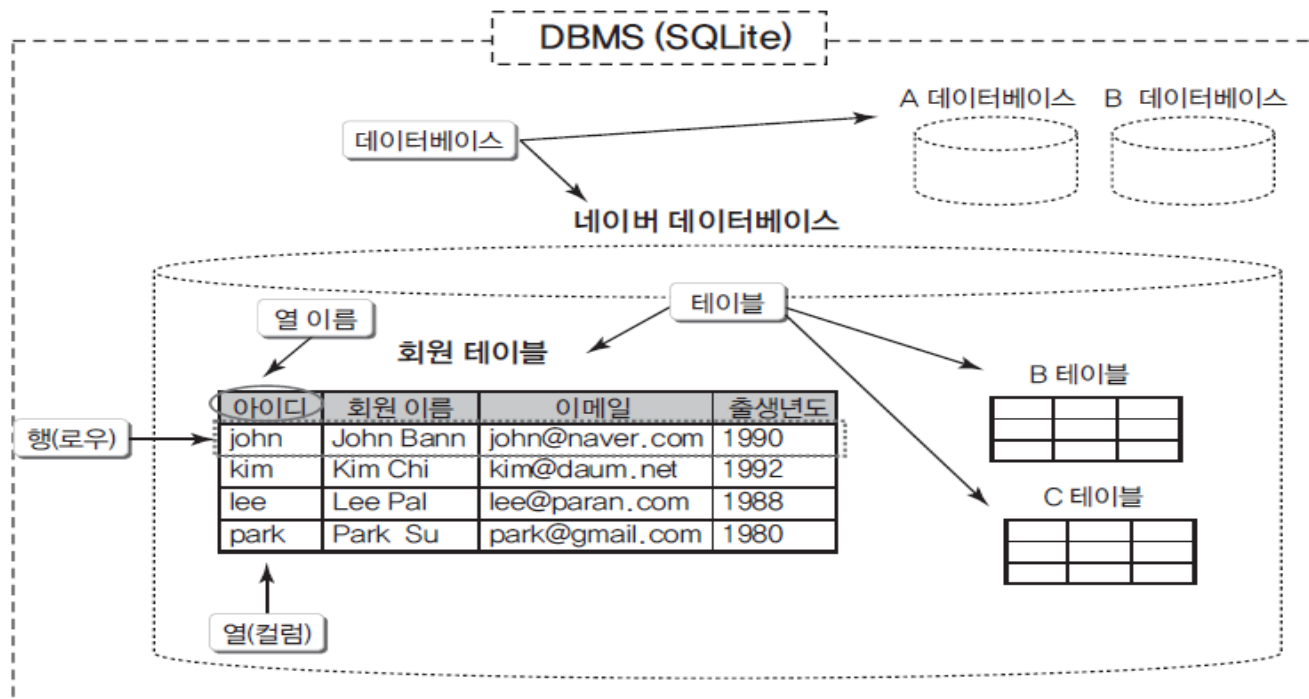


안드로이드 내장 DB

Room 라이브러리

안드로이드 SQLite 내장 DBMS

- ❖ 모바일환경에 최적화된 데이터베이스로, 기본적인 개념과 동작은 PC 용 데이터베이스와 거의 동일함
- ✓ 저장 경로 : data/data/[package_name]/database



Room 라이브러리

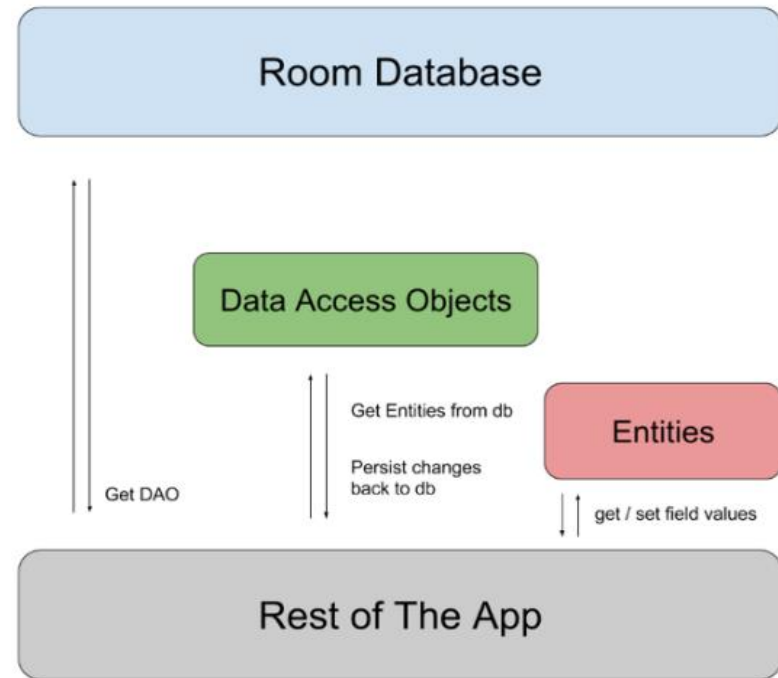
- Room Database / Room DB
 - SQLite에 추상화 계층을 추가하여 데이터베이스에 대한 액세스를 단순화시킨 라이브러리
 - <https://developer.android.com/training/data-storage/room?hl=ko>
 - Room의 이점
 - 런타임시 SQL 쿼리의 유효성 검사
 - 반복적이고 오류가 발생하기 쉬운 상용구 코드를 최소화하는 편리한 어노테이션 기반 처리
 - 비동기식으로 데이터베이스 작업을 처리함
 - build.gradle(Module:app)에 라이브러리 추가하기

'androidx.room:room-ktx'

'androidx.room:room-compiler'

Room 라이브러리

- 기본 구성요소
 - Database
 - 앱에 저장되어 있는 로컬 데이터에 대한 액세스 포인트 제공
 - DAO (Data Access Objects)
 - 데이터베이스에 Insert, Delete, Update 작업할 수 있는 메소드 제공
 - Entity
 - 데이터베이스 내에 존재하는 테이블



Entity 만들기

- 데이터베이스의 테이블 생성

```
@Entity
data class User(
    @PrimaryKey val uid: Int,
    @ColumnInfo(name = "first_name") val firstName: String?,
    @ColumnInfo(name = "last_name") val lastName: String?
)
```

- @Entity : 테이블명 @Entity(tableName = "UserTable")
- @PrimaryKey : 기본키 @PrimaryKey(autoGenerate = true)
- @ColumnInfo : 속성이름

DAO 만들기

- 테이블의 데이터와 상호작용하는 데 필요한 메소드 제공

```
@Dao
interface UserDao {
    @Query("SELECT * FROM user")
    fun getAll(): List<User>

    @Query("SELECT * FROM user WHERE uid IN (:userIds)")
    fun loadAllByIds(userIds: IntArray): List<User>

    @Query("SELECT * FROM user WHERE first_name LIKE :first AND " +
        "last_name LIKE :last LIMIT 1")
    fun findByName(first: String, last: String): User

    @Insert
    fun insertAll(vararg users: User)

    @Delete
    fun delete(user: User)
}
```

@Insert(onConflict = OnConflictStrategy.**IGNORE**)

@Insert(onConflict = OnConflictStrategy.**REPLACE**)

Database 만들기

- 데이터베이스를 보유할 데이터베이스 클래스

```
@Database(entities = [User::class], version = 1)
abstract class AppDatabase : RoomDatabase() {
    abstract fun userDao(): UserDao
}
```

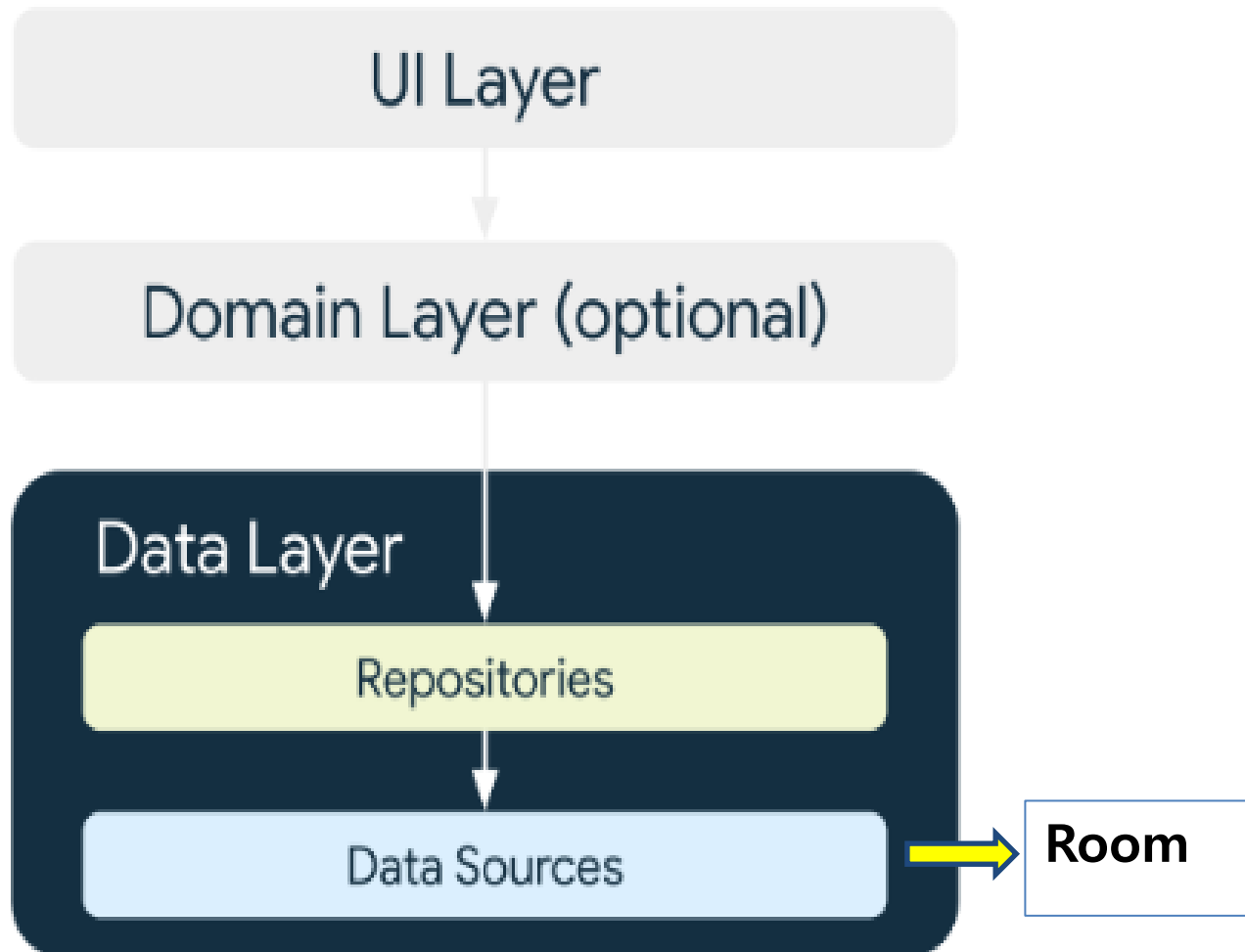
- 데이터베이스 인스턴스 생성

```
val db = Room.databaseBuilder(
    applicationContext,
    AppDatabase::class.java, "database-name"
).build()
```

- Dao 메소드 호출

```
val userDao = db.userDao()
val users: List<User> = userDao.getAll()
```

Room 이용시 권장 아키텍처




Room 관련 라이브러리 확인 및 설정

- App의 build.gradle

```
plugins { this: PluginDependenciesSpecScope
    alias(libs.plugins.androidApplication)
    alias(libs.plugins.jetbrainsKotlinAndroid)
    kotlin("kapt")
}
```

```
dependencies { this: DependencyHandlerScope

    //Room
    implementation(libs.androidx.room.ktx)
    kapt(libs.androidx.room.compiler)
}

kapt { this: KaptExtension
     correctErrorTypes=true
}
```

Kotlin Symbol Processing(KSP) 설정

- KSP
 - Kotlin 코드에서 Java 어노테이션 사용할 수 있는 기능 제공
 - Kapt 보다 빌드 시간이 최대 2배 빠름
- 라이브러리 설치 및 설정

"com.google.devtools.ksp"

최신버전 : "2.0.0-RC3-1.0.20"

- libs.versions.toml
 - 라이브러리 의존성 버전 관리

```
[versions]
symbolProcessing = "2.0.0-RC3-1.0.20"

[plugins]
androidApplication = { id = "com.android.application", version.ref = "agp" }
jetbrainsKotlinAndroid = { id = "org.jetbrains.kotlin.android", version.ref = "kotlin" }
compose-compiler = { id = "org.jetbrains.kotlin.plugin.compose", version.ref = "kotlin" }
```

Kotlin Symbol Processing(KSP) 설정

- Build.gradle (Project수준)

```
plugins { this: PluginDependenciesSpecScope
    alias(libs.plugins.androidApplication) apply false
    alias(libs.plugins.jetbrainsKotlinAndroid) apply false
    alias(libs.plugins.compose.compiler) apply false
    id("com.google.devtools.ksp") version "2.0.0-RC3-1.0.20" apply false
}
```

- Build.gradle (app 수준)

```
plugins { this: PluginDependenciesSpecScope
    alias(libs.plugins.androidApplication)
    alias(libs.plugins.jetbrainsKotlinAndroid)
    alias(libs.plugins.compose.compiler)
    id("com.google.devtools.ksp")
}
```

```
dependencies { this: DependencyHandlerScope
    //Room
    implementation(libs.androidx.room.ktx)
    implementation(libs.symbol.processing)
    ksp(libs.androidx.room.compiler)
}
```

예제. CRUD

- 데이터베이스에 삽입, 수정, 삭제, 검색 예제

Item ID
Item Name
Item Quantity
<div>Insert</div> <div>Update</div> <div>delete</div> <div>find</div>
1 greenjoa 10
2 greenjoa1 50
3 greenjoa2 10

수고하셨습니다.