Class 7: Local Persistence (Room Database)

Goal: Saving data efficiently on the device

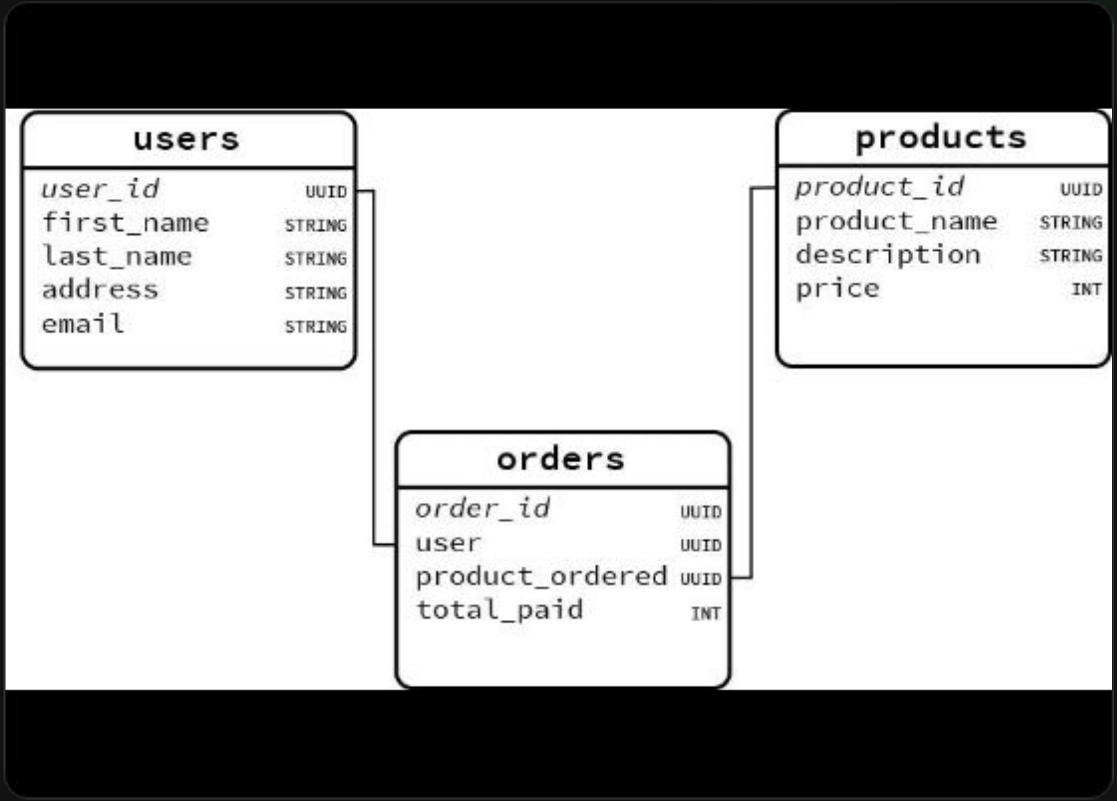
Instructor: Mark Joseli

The Goal

To understand how to persist data locally on an Android device efficiently using modern Architecture Components, ensuring a robust offline-first experience.

Theory: SQL Basics (CRUD)

- **Create (Insert):** Adding new records to the database tables.
- **Read (Select):** Querying the database to retrieve specific data points.
- **Update:** Modifying existing records to reflect changes.
- **Delete:** Removing data that is no longer needed or valid.



What is Room Database?

Abstraction Layer

Room provides an abstraction layer over SQLite to allow fluent database access while harnessing the full power of SQLite. It reduces boilerplate code significantly compared to raw SQLite.

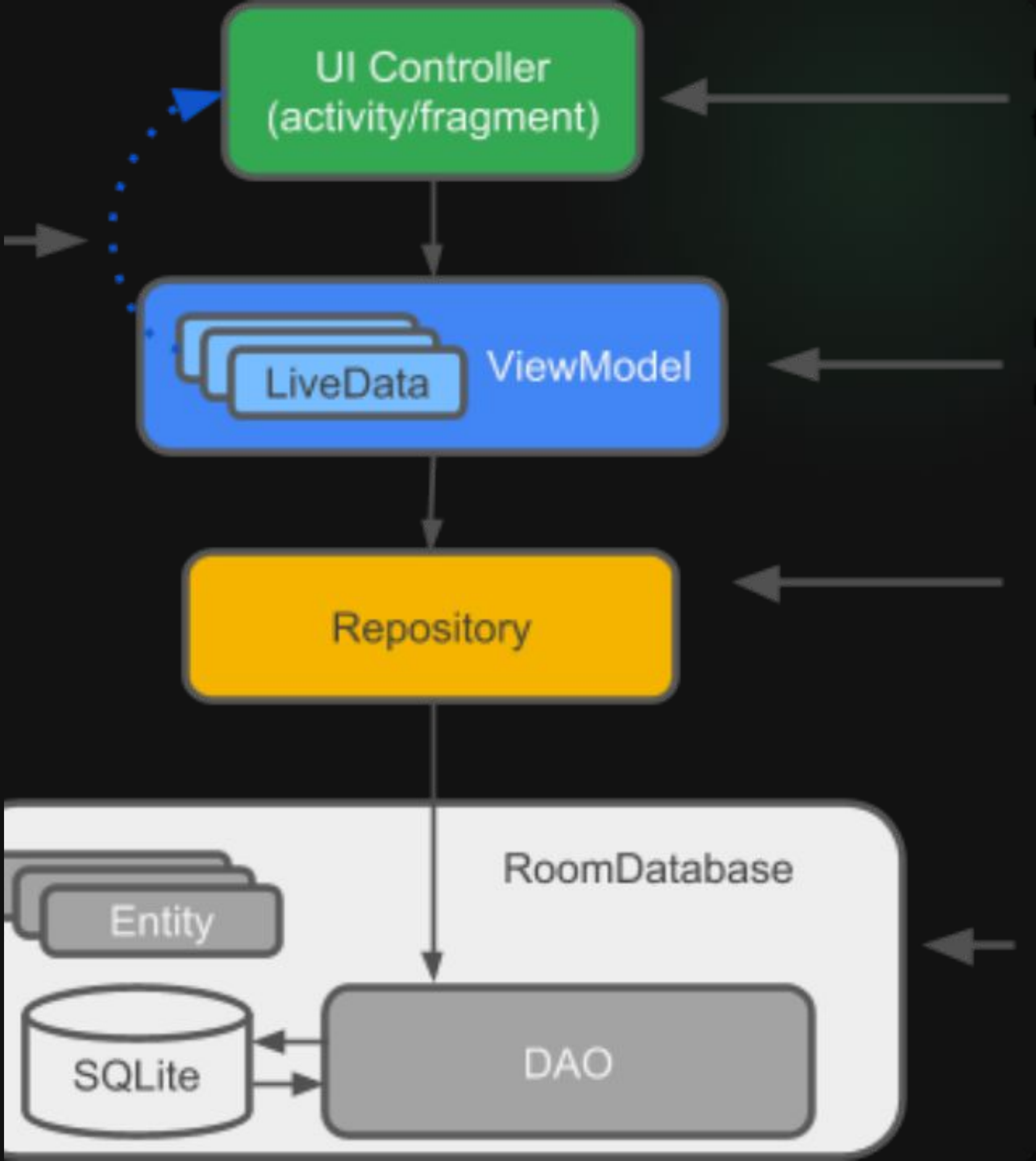
Compile-Time Checks

One of Room's strongest features is its ability to validate your SQL queries at compile-time. If you misspell a table name or column, the build fails immediately, preventing runtime crashes.

Room Architecture

The Three Major Components

- **1. Database:** Contains the database holder and serves as the main access point for the underlying connection.
- **2. Entity:** Represents a table within the database. Each instance represents a row.
- **3. DAO (Data Access Object):** Contains the methods used for accessing the database (queries, inserts, etc.).



Component Breakdown

Entity

Annotated class that describes a database table when working with Room.

DAO

Interface defining methods to access the database. Maps method calls to SQL queries.

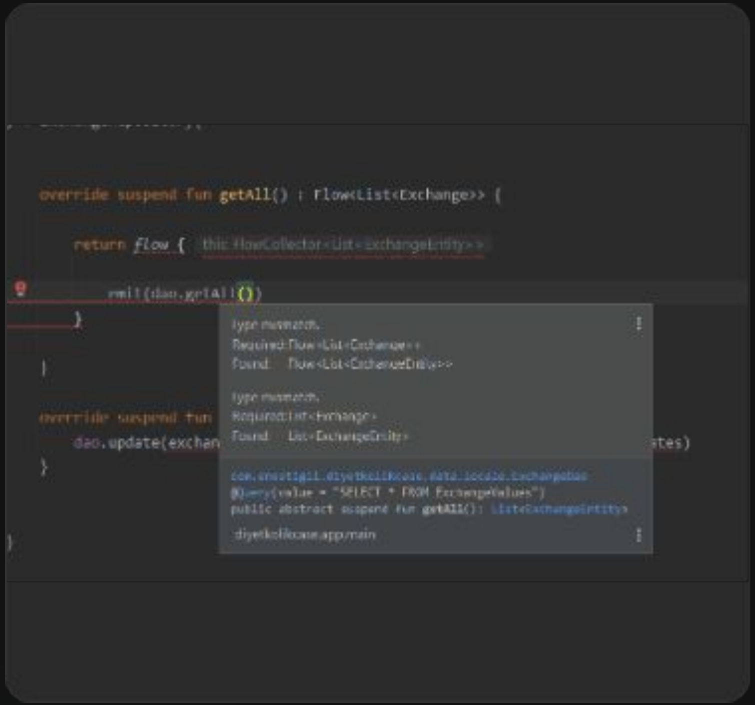
Repository

A class that abstracts access to multiple data sources (Room + Network).

Why use a Repository?

- **Decoupling** It decouples the application (UI/ViewModel) from the data sources.
- **Clean API** Provides a clean, simple API for data access to the rest of the application.
- **Single Source of Truth** Manages multiple backends (e.g., fetching from API, saving to Room) seamlessly.
- **Testing** Makes the code much easier to test by allowing you to mock the data layer.

Implementation Snippets



```
    override suspend fun getAll() : Flow<List<Exchange>> {

        return flow {  this: FlowCollector<List<ExchangeEntity>>

            emit(dao.getAll())
        }

    }

    override suspend fun
        dao.update(exchan                                    tes)
    }

}
```

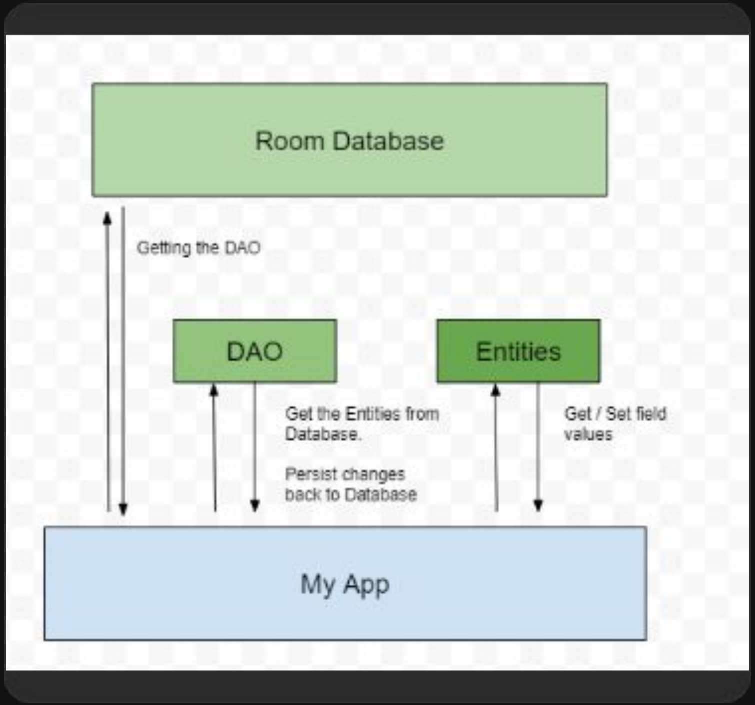
```
Type mismatch.
Required: Flow<List<Exchange>>
Found:    Flow<List<ExchangeEntity>>

Type mismatch.
Required: List<Exchange>
Found:    List<ExchangeEntity>

com.[illegible].diyetkolikcase.data.locale.ExchangeDao
@Query(value = "SELECT * FROM ExchangeValues")
public abstract suspend fun getAll(): List<ExchangeEntity>

diyetkolikcase.app.main
```

@Entity Data Class



@Dao Interface

Kotlin code snippet of the RoomDatabase abstract class

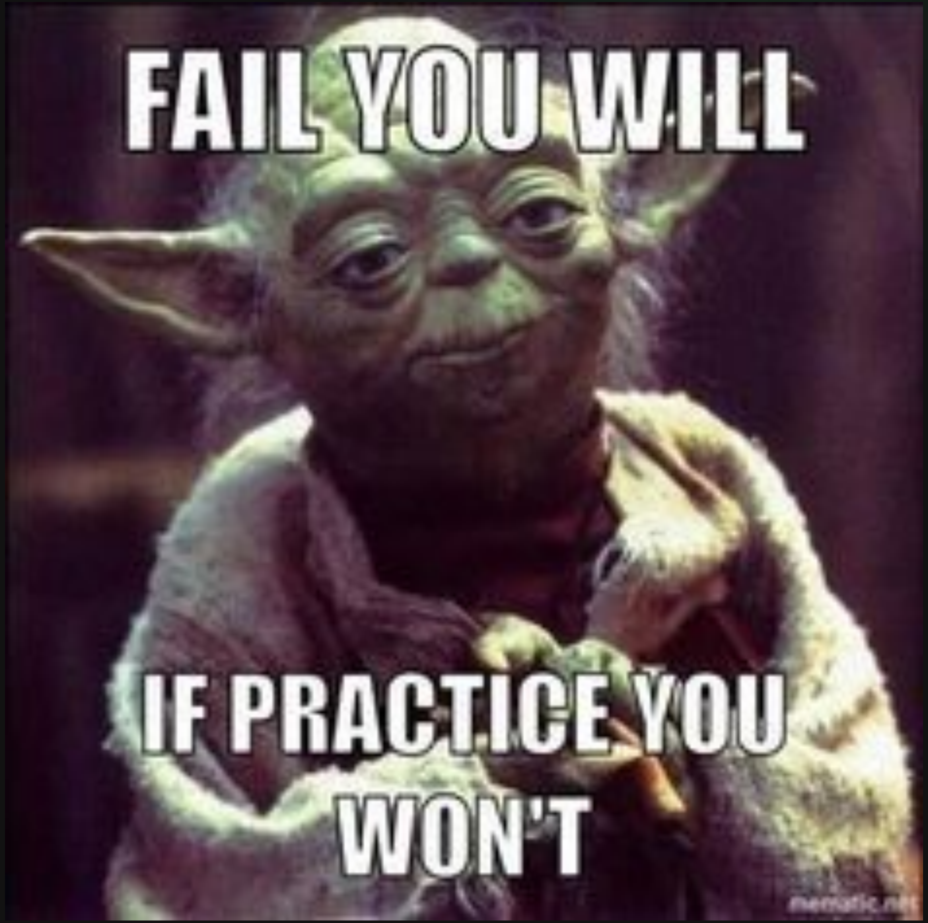
@Database Class

Study Resources

"Practice makes perfect. Since we don't have a lab today, use these resources to build your own implementation."

- ✓ **Official CodeLab**
  https://developer.android.com/codelabs/basic-android-kotlin-compose-persisting-data-room

- ✓ **Video Tutorial** "Android Jetpack: Room":
  https://www.youtube.com/watch?v=te_UGGHWMeI&t=5s



Questions?

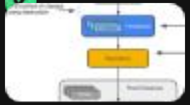
Thank you for attending Class 7.

✉ mark.joseli@pucpr.br

Image Sources

https://images.ctfassets.net/00voh0j35590/174aQuVktxmYMgoEP4sUsj/adb42dd2494ef66edcab3a4286d2b1af/database-schema-example-diagram.jpg

Source: www.cockroachlabs.com



https://google-developer-training.github.io/android-developer-fundamentals-course-concepts-v2/images/10-1-c-room-livedata-viewmodel/dg_architecture_comonents.png

Source: google-developer-training.github.io

https://i.sstatic.net/oZ5WS.png

Source: stackoverflow.com

https://miro.medium.com/1*XqWEHZqK8vkoBAIhygaIBA.png

Source: medium.com

https://images.hive.blog/DQmSZyxgB8zn7nz2LfWPQrRh4JQVM19VEx5XPW3xiCGfYhw/carbon%20(3).png

Source: hive.blog

https://img.freepik.com/premium-vector/boy-student-learning-laptop-data-analysis-coding-school-child-computer-studying-programming-information-technology-online-education-fla
t-vector-illustration-isolated-white-background_198278-27771.jpg

Source: www.freepik.com