Module 5 : Data Storage and Persistence in Android

Differences Between SharedPreferences, SQLite, and Room in Android

---

1. SharedPreferences

What is it? SharedPreferences is a lightweight mechanism to store and retrieve key-value pairs of primitive data types (such as integers, booleans, and strings). It is typically used for saving small amounts of data that need to persist across user sessions.

How it works:

- Data is stored in XML files within the app's private storage.
- Provides methods to save, retrieve, and modify values associated with specific keys.

Use Cases:

- Storing user preferences, such as theme selection or notification settings.
- Maintaining app state flags, such as whether the user has seen a tutorial.
- Caching small amounts of non-critical data, like the last username entered.

Pros:

- Easy to implement with a simple API.
- Lightweight and optimized for small datasets.
- No need for external libraries or complex setups.

Cons:

- Not suitable for storing large or complex data.
- Limited to basic key-value pairs; no relational or query support.
- Can cause performance issues if misused on the main thread for frequent writes.

---

2. SQLite

What is it? SQLite is a full-featured relational database engine that is built into Android. It allows developers to store, query, and manage structured data with relationships between different entities.

How it works:

- Data is stored in tables, with rows representing records and columns defining attributes.
- SQL queries are used to perform CRUD (Create, Read, Update, Delete) operations.
- Developers are responsible for creating and maintaining the database schema.

Use Cases:

- Managing structured data for applications, such as user profiles or product catalogs.
- Offline storage for apps that need to sync data with a remote server.
- Storing large datasets or data with complex relationships.

Pros:

- Offers advanced querying capabilities using SQL.
- High performance for large datasets.
- Fully customizable database schema.

Cons:

- Requires manual SQL coding, which can lead to errors and maintenance challenges.
- Schema migrations can be tedious and error-prone.
- Increased complexity compared to simpler storage options.

---

3. Room

What is it? Room is a persistence library provided by Android Jetpack that acts as an abstraction layer over SQLite. It simplifies database management and reduces boilerplate code, making it easier to work with relational data.

How it works:

- Uses annotations to define database tables, columns, and relationships.
- Data access is managed through Data Access Objects (DAOs), which provide compile-time query verification.
- Supports LiveData and Flow for observing database changes in a reactive manner.
- Handles schema migrations automatically through annotations and helper methods.

Use Cases:

- Storing structured data with complex relationships in modern apps.
- Integrating databases with Android Jetpack components, such as ViewModel and LiveData.
- Replacing SQLite for new projects to reduce development complexity.

Pros:

- Simplifies database operations and reduces boilerplate code.
- Provides compile-time validation of SQL queries, reducing runtime errors.
- Integrates seamlessly with other Jetpack components.
- Offers tools for automated schema migrations.

Cons:

- May introduce slight performance overhead compared to raw SQLite for very simple use cases.
- Requires learning annotations and DAO patterns, which may have a learning curve for beginners.

When to Choose

1. **SharedPreferences:**
   - Use when you need to store small amounts of simple data, such as user preferences or flags.
   - Best for lightweight use cases where no complex relationships or queries are needed.
2. **SQLite:**
   - Ideal for managing large datasets or data with complex relationships.
   - Suitable for apps requiring custom SQL queries or advanced database operations.
   - Prefer when fine-grained control over the database schema is required.
3. **Room:**
   - The best choice for most modern apps requiring relational database storage.
   - Use when you want an easier and safer way to manage databases.
   - Recommended for projects leveraging Android Jetpack and reactive programming.