

Discuss the differences between SharedPreferences, SQLite, and Room in Android. When would you choose each for data storage

1. SharedPreferences

- **Description:** SharedPreferences is used to store small amounts of primitive data (key-value pairs).
- **Data Type:** Supports primitive data types like `String`, `int`, `boolean`, `float`, `long`, and `Set<String>`.
- **Use Cases:**
 - Storing user preferences/settings (e.g., dark mode, notification settings).
 - Session management (e.g., saving login status, user token).
 - Small configuration data.
- **Advantages:**
 - Simple and easy to use.
 - Persistent across app restarts.
- **Disadvantages:**
 - Not suitable for complex or large data.
 - No querying support.

2. SQLite

- **Description:** SQLite is a relational database for Android that provides structured data storage.
- **Data Type:** Supports multiple data types including `TEXT`, `INTEGER`, `REAL`, and `BLOB`.
- **Use Cases:**
 - Storing structured data with relationships (e.g., user information, product catalogs).
 - Applications that require advanced queries and data manipulations.
 - Data that needs indexing for performance.
- **Advantages:**
 - Allows complex queries using SQL.
 - Good for moderate to large amounts of structured data.
- **Disadvantages:**
 - Requires manual management of database schema.
 - No compile-time validation, leading to potential runtime errors.
 - Requires more boilerplate code for CRUD operations.

3. Room

- **Description:** Room is a high-level abstraction over SQLite that provides a more convenient, type-safe, and boilerplate-free way of working with databases.
- **Data Type:** Supports multiple data types, similar to SQLite.
- **Use Cases:**
 - Structured data with relationships, but with less boilerplate.
 - Apps that need offline-first functionality.
 - Storing large amounts of complex data.
- **Advantages:**
 - Simplified code with annotations for entities, DAOs (Data Access Objects), and relationships.
 - Compile-time verification of SQL queries and schema.
 - Built-in support for LiveData and Flow.
 - Provides migration support for database schema changes.
- **Disadvantages:**
 - Slightly more complex to set up compared to SharedPreferences.
 - Adds an abstraction layer over SQLite, which may add overhead for very simple cases.