Note-Taking app using SQLite (SQLiteOpenHelper) plus RecyclerView and full CRUD (Create, Read, Update, Delete).

Package used: com.example.notesapp — make sure your project package matches this (or change the package lines to your package).

Important files:

- **DBHelper.kt** — SQLiteOpenHelper + CRUD methods
- **Note.kt** — data model
- **NoteAdapter.kt** — RecyclerView adapter with click handlers
- **MainActivity.kt** — shows list of notes, delete, and open Add/Edit screen
- **AddEditNoteActivity.kt** — add or edit a note
- **Layouts**: activity_main.xml, item_note.xml, activity_add_edit.xml
- **AndroidManifest.xml** entries and required Gradle dependencies

Note:
Follow the file-by-file code below. Put each file in the indicated location.

1) AndroidManifest.xml (ensure activities declared)
app/src/main/AndroidManifest.xml — keep other existing lines; replace or merge application section:

```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.notesapp">

  <application
    android:allowBackup="true"
    android:label="NotesApp"
    android:icon="@mipmap/ic_launcher"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:theme="@style/Theme.AppCompat.Light.DarkActionBar">

    <activity android:name=".AddEditNoteActivity" />
    <activity android:name=".MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>

  </application>

</manifest>
```

2) Gradle dependencies (app-level)

Open app/build.gradle and inside dependencies { ... } add:

```
implementation 'androidx.recyclerview:recyclerview:1.3.2'
implementation 'com.google.android.material:material:1.9.0'
implementation 'androidx.appcompat:appcompat:1.6.1'
implementation 'androidx.core:core-ktx:1.10.1'
```

Also ensure compileSdk and targetSdk are appropriate and (optionally) enable viewBinding if you prefer. Not required for this code.

After editing, click **Sync Now**.

3) Layouts

a) activity_main.xml

app/src/main/res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  android:id="@+id/coordinator"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvNotes"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="8dp"
    android:clipToPadding="false" />

  <com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fabAdd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:contentDescription="Add note"
    app:srcCompat="@android:drawable/ic_input_add"
    app:tint="@android:color/white"
    app:layout_anchor="@id/rvNotes"
    app:layout_anchorGravity="bottom|end"
    android:layout_margin="16dp"/>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

b) item_note.xml

app/src/main/res/layout/item_note.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:card="http://schemas.android.com/apk/res-auto"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  card:cardElevation="4dp"
  card:cardCornerRadius="6dp"
  android:layout_margin="6dp">

  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="12dp"
    android:orientation="vertical">

    <TextView
      android:id="@+id/tvTitle"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:textStyle="bold"
      android:textSize="18sp"
      android:text="Note title" />

    <TextView
      android:id="@+id/tvContent"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="6dp"
      android:text="Note content"
      android:textSize="14sp" />

    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="horizontal"
      android:gravity="end"
      android:layout_marginTop="8dp">

      <TextView
        android:id="@+id/tvDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="12sp"
        android:textColor="#666666" />
    </LinearLayout>
  </LinearLayout>
</androidx.cardview.widget.CardView>
```

c) activity_add_edit.xml

app/src/main/res/layout/activity_add_edit.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
  android:fillViewport="true"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:padding="16dp">

  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <EditText
      android:id="@+id/etTitle"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Title" />

    <EditText
      android:id="@+id/etContent"
      android:layout_width="match_parent"
      android:layout_height="200dp"
      android:gravity="top"
      android:hint="Write your note here"
      android:inputType="textMultiLine"
      android:layout_marginTop="12dp" />

    <Button
      android:id="@+id/btnSave"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Save"
      android:layout_marginTop="16dp" />

  </LinearLayout>
</ScrollView>
```

4) Kotlin files

Place these under app/src/main/java/com/example/notesapp/

a) Note.kt (model)

```
package com.example.notesapp

data class Note(
    val id: Int = 0,
    val title: String,
    val content: String,
    val timestamp: Long = System.currentTimeMillis()
)
```

b) DBHelper.kt (SQLiteOpenHelper + CRUD)

```
package com.example.notesapp

import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class DBHelper(context: Context) : SQLiteOpenHelper(context, DB_NAME, null, DB_VERSION) {

    companion object {
        private const val DB_NAME = "notes_db"
        private const val DB_VERSION = 1
        private const val TABLE = "notes"
        private const val COL_ID = "id"
        private const val COL_TITLE = "title"
        private const val COL_CONTENT = "content"
        private const val COL_TIMESTAMP = "timestamp"
    }

    override fun onCreate(db: SQLiteDatabase) {
        val sql = """
            CREATE TABLE $TABLE (
                $COL_ID INTEGER PRIMARY KEY AUTOINCREMENT,
                $COL_TITLE TEXT,
                $COL_CONTENT TEXT,
                $COL_TIMESTAMP INTEGER
            )
        """.trimIndent()
        db.execSQL(sql)
    }
    override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {
        db.execSQL("DROP TABLE IF EXISTS $TABLE")
        onCreate(db)
    }

    fun addNote(title: String, content: String): Long {
        val db = writableDatabase
        val cv = ContentValues()
        cv.put(COL_TITLE, title)
        cv.put(COL_CONTENT, content)
        cv.put(COL_TIMESTAMP, System.currentTimeMillis())
        return db.insert(TABLE, null, cv)
    }

    fun getAllNotes(): ArrayList<Note> {
        val list = ArrayList<Note>()
        val db = readableDatabase
        val cursor = db.rawQuery("SELECT * FROM $TABLE ORDER BY $COL_TIMESTAMP DESC", null)
        if (cursor.moveToFirst()) {
            do {
                val note = Note(
                    id = cursor.getInt(cursor.getColumnIndexOrThrow(COL_ID)),
                    title = cursor.getString(cursor.getColumnIndexOrThrow(COL_TITLE)),
                    content = cursor.getString(cursor.getColumnIndexOrThrow(COL_CONTENT)),
                    timestamp = cursor.getLong(cursor.getColumnIndexOrThrow(COL_TIMESTAMP))
                )
                list.add(note)
            } while (cursor.moveToNext())
        }
        cursor.close()
        return list
    }

    fun getNoteById(id: Int): Note? {
        val db = readableDatabase
        val cursor = db.rawQuery("SELECT * FROM $TABLE WHERE $COL_ID = ?",
arrayOf(id.toString()))
        var note: Note? = null
        if (cursor.moveToFirst()) {
            note = Note(
                id = cursor.getInt(cursor.getColumnIndexOrThrow(COL_ID)),
                title = cursor.getString(cursor.getColumnIndexOrThrow(COL_TITLE)),
                content = cursor.getString(cursor.getColumnIndexOrThrow(COL_CONTENT)),
                timestamp = cursor.getLong(cursor.getColumnIndexOrThrow(COL_TIMESTAMP))
            )
        }
        cursor.close()
        return note
    }

    fun updateNote(id: Int, title: String, content: String): Int {
        val db = writableDatabase
        val cv = ContentValues()
        cv.put(COL_TITLE, title)
        cv.put(COL_CONTENT, content)
        cv.put(COL_TIMESTAMP, System.currentTimeMillis())
        return db.update(TABLE, cv, "$COL_ID = ?", arrayOf(id.toString()))
    }

    fun deleteNote(id: Int) {
        val db = writableDatabase
        db.delete(TABLE, "$COL_ID = ?", arrayOf(id.toString()))
    }
}
```

c) NoteAdapter.kt (RecyclerView Adapter)

```
package com.example.notesapp

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class NoteAdapter(
    private val notes: ArrayList<Note>,
    private val onItemClick: (Note) -> Unit,
    private val onItemLongClick: (Note) -> Unit
) : RecyclerView.Adapter<NoteAdapter.NoteVH>() {

    inner class NoteVH(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val tvTitle: TextView = itemView.findViewById(R.id.tvTitle)
        val tvContent: TextView = itemView.findViewById(R.id.tvContent)
        val tvDate: TextView = itemView.findViewById(R.id.tvDate)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): NoteVH {
        val view = LayoutInflater.from(parent.context).inflate(R.layout.item_note, parent, false)
        return NoteVH(view)
    }

    override fun onBindViewHolder(holder: NoteVH, position: Int) {
        val note = notes[position]
        holder.tvTitle.text = note.title
    // show a short preview of content (first 100 chars)
    holder.tvContent.text = if (note.content.length > 200) note.content.substring(0, 200) + "..."
else note.content
    holder.tvDate.text = android.text.format.DateFormat.format("dd MMM yyyy, hh:mm a",
note.timestamp)

    holder.itemView.setOnClickListener { onItemClick(note) }
    holder.itemView.setOnLongClickListener {
      onItemLongClick(note)
      true
    }
  }

  override fun getItemCount(): Int = notes.size
}
```

d) MainActivity.kt (list, delete, open add/edit)

```
package com.example.notesapp

import android.app.AlertDialog
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.google.android.material.floatingactionbutton.FloatingActionButton

class MainActivity : AppCompatActivity() {

  private lateinit var db: DBHelper
  private lateinit var rvNotes: RecyclerView
  private lateinit var fabAdd: FloatingActionButton
  private val notesList = ArrayList<Note>()
  private lateinit var adapter: NoteAdapter

  companion object {
    const val EXTRA_NOTE_ID = "note_id"
  }

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    db = DBHelper(this)
    rvNotes = findViewById(R.id.rvNotes)
```

```
        fabAdd = findViewById(R.id.fabAdd)

        adapter = NoteAdapter(notesList, onItemClick = { note ->
            // open for edit
            val intent = Intent(this, AddEditNoteActivity::class.java)
            intent.putExtra(EXTRA_NOTE_ID, note.id)
            startActivity(intent)
        }, onItemLongClick = { note ->
            // show delete confirm
            AlertDialog.Builder(this)
                .setTitle("Delete note")
                .setMessage("Delete \"${note.title}\"?")
                .setPositiveButton("Delete") { _, _ ->
                    db.deleteNote(note.id)
                    loadNotes()
                    Toast.makeText(this, "Deleted", Toast.LENGTH_SHORT).show()
                }
                .setNegativeButton("Cancel", null)
                .show()
        })

        rvNotes.layoutManager = LinearLayoutManager(this)
        rvNotes.adapter = adapter

        fabAdd.setOnClickListener {
            val intent = Intent(this, AddEditNoteActivity::class.java)
            startActivity(intent)
        }
    }

    override fun onResume() {
        super.onResume()
        loadNotes()
    }

    private fun loadNotes() {
        notesList.clear()
        notesList.addAll(db.getAllNotes())
        adapter.notifyDataSetChanged()
    }
}
```

e) AddEditNoteActivity.kt (Add / Edit screen)

```kotlin
package com.example.notesapp

import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class AddEditNoteActivity : AppCompatActivity() {

    private lateinit var etTitle: EditText
    private lateinit var etContent: EditText
    private lateinit var btnSave: Button
    private lateinit var db: DBHelper
    private var editingNoteId: Int = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_add_edit)

        etTitle = findViewById(R.id.etTitle)
        etContent = findViewById(R.id.etContent)
        btnSave = findViewById(R.id.btnSave)
        db = DBHelper(this)

        // If an id was passed, we are editing
        editingNoteId = intent.getIntExtra(MainActivity.EXTRA_NOTE_ID, 0)
        if (editingNoteId != 0) {
            val note = db.getNoteById(editingNoteId)
            note?.let {
                etTitle.setText(it.title)
                etContent.setText(it.content)
            }
        }

        btnSave.setOnClickListener {
            val title = etTitle.text.toString().trim()
            val content = etContent.text.toString().trim()

            if (title.isEmpty()) {
                etTitle.error = "Enter title"
                return@setOnClickListener
            }

            if (editingNoteId == 0) {
                val id = db.addNote(title, content)
                if (id > 0) {
                    Toast.makeText(this, "Note added", Toast.LENGTH_SHORT).show()
                    finish()
                } else {
                    Toast.makeText(this, "Error adding note", Toast.LENGTH_SHORT).show()
                }
            } else {
                val rows = db.updateNote(editingNoteId, title, content)
                if (rows > 0) {
                    Toast.makeText(this, "Note updated", Toast.LENGTH_SHORT).show()
                    finish()
                } else {
                    Toast.makeText(this, "Error updating note", Toast.LENGTH_SHORT).show()
                }
            }
        }
    }
}
```

5) Run instructions / Checklist

1. Create a new Android Studio project (or use existing). Ensure package is com.example.notesapp or update package lines accordingly.
2. Add the provided Gradle dependencies and sync.
3. Add the layout XML files in res/layout/.
4. Add Kotlin files under app/src/main/java/com/example/notesapp/.
5. Ensure AndroidManifest.xml includes both activities (shown above).
6. Clean/Build the project, then Run on emulator/device.

6) Notes:

- DBHelper uses SQLiteOpenHelper — **onCreate()** creates the table once.
- **getAllNotes()** returns results ordered by newest first.
- MainActivity refreshes the list in **onResume()** so changes (add/edit/delete) are reflected.
- AddEditNoteActivity detects whether it's adding or editing by checking the EXTRA_NOTE_ID.
- Long-press on a note to delete (confirm dialog). Tap to edit.