1. To **Change the Size of ImageView**:

Ex: in the XML File You Add Under the ImageView

<ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/background"  
 android:scaleType="fitXY"/>

**Note**: This Will Make **the Image Expand** to **Take All the Screen**

EX: to make the image **Take All the Screen** **Without Expanding**

<ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:scaleType="fitXY"  
 android:adjustViewBounds="true"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/art\_background" />

**Note**: For **More Ways** Visit this [WebSite](https://guides.codepath.com/android/Working-with-the-ImageView)

1. **Save in SQLite**:
2. Create **SQLHelper class**:

class DBHelper(  
 context: Context?,  
 name: String?= "details.db",  
 factory: SQLiteDatabase.CursorFactory?= null,  
 version: Int= 1,  
 private val tableName: String= "notes"  
)

: SQLiteOpenHelper(context, name, factory, version) {  
 override fun onCreate(p0: SQLiteDatabase?) {  
 p0?.execSQL("create table $tableName (Note Text)")  
 }  
  
 override fun onUpgrade(p0: SQLiteDatabase?, p1: Int, p2: Int) {}  
  
 fun saveNotes(note: String): Long {  
 val cv= ContentValues()  
 cv.put("Note",note)  
 return *writableDatabase*.insert(tableName,null,cv)  
 }  
}

1. **In Main Class**:

val dbHelper= DBHelper(this)

val check= dbHelper.saveNotes(noteEntry.*text*.toString())  
val wrongCode: Long= -1  
if (check != wrongCode) {  
 StyleableToast.makeText(this, "Saved Successfully!!\n$check", R.style.*mytoast*).show()  
}  
else  
 StyleableToast.makeText(this,"Something Went Wrong!!\n$check",R.style.*mytoast*).show()

1. The **Deference** between (**this** and **applicationContext**) when we need to use Context:

If we are going to use it **inside the same activity** it’s ok to use (**this**)

**But** if we need to **send it to another function** or **another class**, better to use (**applicationContext**).

Because if we pass the activity context and that activity got destroyed the **app will crash**.