***USING ROOM***

There are majorly 3 major components of room

1. **Entities** – It represents the table within the database
2. **DAO(Data access objects)** – It contains the methods used for accessing the database.
3. **Database** – It contains the database holder and serves as the main access point for the connection to your app’s data. Therefore the class annotated with the @Database should be an abstract class which extends Room Database.
4. It should include the list of all the entities(table) associated with that database
5. It should contain the abstract methods that has 0 arguments and return the class that is annotated with @Dao

We usually create the instance of the database mentioned in the 3rd point to access it by calling Room.databaseBuilder()

@Entity(tableName = “Users”)  
data class User(  
    @PrimaryKey val uid: Int,  
    @ColumnInfo(name = "first\_name") val firstName: String?,  
    @ColumnInfo(name = "last\_name") val lastName: String?  
)

@Dao  
interface UserDao {  
    @Query("SELECT \* FROM user")  
    fun getAll(): List<User>  
  
    @Query("SELECT \* FROM user WHERE uid IN (:userIds)")  
    fun loadAllByIds(userIds: IntArray): List<User>  
  
    @Query("SELECT \* FROM user WHERE first\_name LIKE :first AND " +  
           "last\_name LIKE :last LIMIT 1")  
    fun findByName(first: String, last: String): User  
  
    @Insert  
    fun insertAll(vararg users: User)  
  
    @Delete  
    fun delete(user: User)  
}

@Database(entities = arrayOf(User::class), version = 1)  
abstract class AppDatabase : RoomDatabase() {  
    abstract fun userDao(): UserDao  
}

val db = Room.databaseBuilder(  
            applicationContext,  
            AppDatabase::class.java, "database-name"  
        ).build()

***Variations In making Entity***

Each entity must define at least 1 field as a primary key. Even when there is only 1 field, you still need to annotate the field with the [@PrimaryKey](https://developer.android.com/reference/androidx/room/PrimaryKey.html) annotation. Also, if you want Room to assign automatic IDs to entities, you can set the @Primarykey’s auto generation property as true

By default if no name is mentioned specifically to the table name then the table takes the name of the class name

Like for below table , table name is “User” where the column name is id,firstName,lastName

**@Entity**  
data class User(  
    @PrimaryKey var id: Int,  
    var firstName: String?,  
    var lastName: String?  
)