```
--- TrackDatabase.kt ---

/**
* [TrackDatabase] defines the database configuration and serves as the app's main access point to
* the persisted data.
*/
@Database(entities = [TrackPoint::class], version = 1)
abstract class TrackDatabase : RoomDatabase() {
   abstract fun trackDao(): TrackDao
}

--- TrackPoint.kt ---

/** Each instance of [TrackPoint] represents a row in a `records` table in the app's database. */
@Entity(tableName = "records", indices = [Index(value = ["trackId"])])
data class TrackPoint(
   val trackId: Int,
   @ColumnInfo(name = "timestamp") val timestamp: Long,
   @ColumnInfo(name = "stroke_rate") val strokeRate: Double,
   @ColumnInfo(name = "latitude") val latitude: Double? = null,
   @ColumnInfo(name = "longitude") val longitude: Double? = null,
   /**
    * The speed of the boat at the given timestamp, in metres per second
    */
   @ColumnInfo(name = "speed") val speed: Float? = null,

   /**
    * Automatically incremented point ID. For simple sequential ordering. It is placed last in the
    * constructor so that it is not necessary to use named arguments when creating a new
    * trackpoint.
    */
   @PrimaryKey(autoGenerate = true) val pointId: Int = 0,
)

--- TrackDao.kt ---

/**
* [TrackDao] provides the methods that the rest of the app uses to interact with data in the
* `tracks` table.
*/
@Dao
interface TrackDao {
   @Query(
        "SELECT trackId, MIN(timestamp) AS timestamp FROM records GROUP BY trackId ORDER BY timestamp
DESC")
   suspend fun getSessions(): List<Session>

   @Query("SELECT MAX(trackId) FROM RECORDS") suspend fun getLastSessionId(): Int?

   @Query("SELECT * FROM records WHERE trackId == :sessionId ORDER BY pointId ASC")
   suspend fun loadSession(sessionId: Int): List<TrackPoint>

   @Insert(onConflict = OnConflictStrategy.ABORT) suspend fun insert(vararg records: TrackPoint)
}

--- Session.kt ---

data class Session(
   val trackId: Int,
   @ColumnInfo(name = "timestamp") val timestamp: Long,
) {
   override fun toString(): String =
       SimpleDateFormat("EEE MMM d HH:mm ''yy", Locale.UK).format(Date(timestamp))
}
```