

Міністерство освіти і науки України

Національний університет «Львівська політехніка»



Лабораторна робота №2

з курсу:

Організація баз даних та знань

Виконав:

студент групи КН-207

Гірняк Т. О.

Прийняла:

Мельникова Н.І.

Львів – 2019 р.

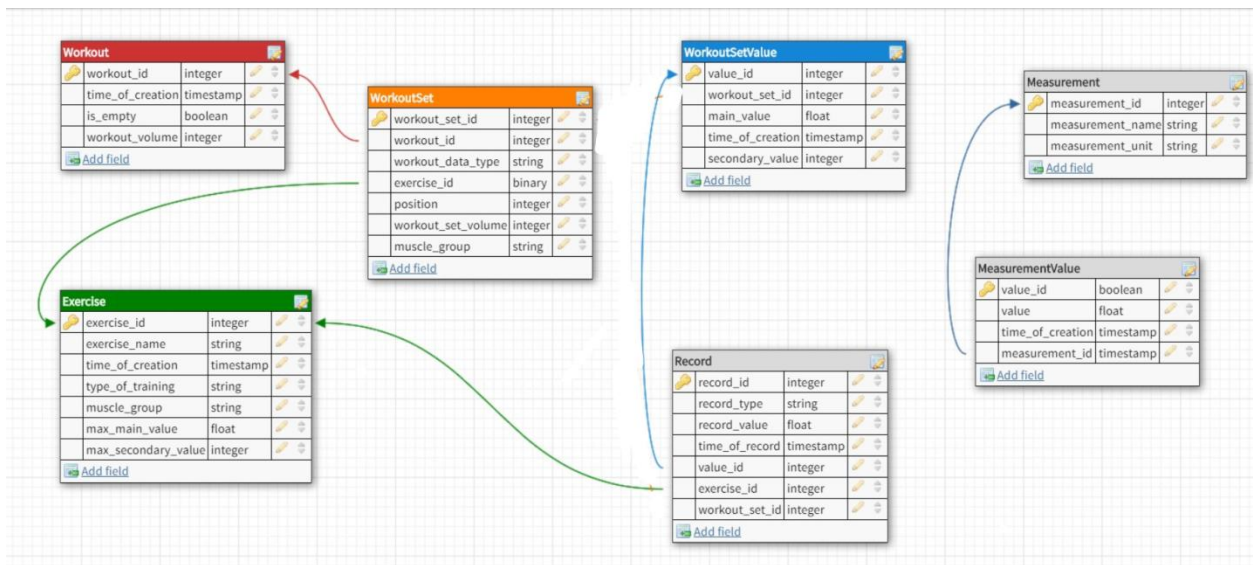
Тема:

Створення таблиць бази даних засобами SQL

Мета:

Побудувати даталогічну модель бази даних; визначити типи, розмірності та обмеження полів; визначити обмеження таблиць; розробити SQL запити для створення спроектованих таблиць.

Модель у вигляді схеми:



Імплементація бази даних у SQLite з використанням мови Kotlin:

```
@Entity
class ExercisesTable(@field:PrimaryKey
    var itemId: Int,
    @field:ColumnInfo
    var muscleGroup: String,
    @field:ColumnInfo
    var exerciseName: String,
    @field:ColumnInfo
    var typeOfTraining: String,
    @field:ColumnInfo
    var workData: String,
    @field:ColumnInfo
    var maxMainValue: Double,
    @field:ColumnInfo
    var maxMainValueId: Int,
    @field:ColumnInfo
    var maxSecondaryValue: Int,
    @field:ColumnInfo
    var maxSecondaryValueId: Int)
```

```

@Entity
class MeasurementItem(@field:PrimaryKey
    var itemId: Int,
    @field:ColumnInfo
    var measurementName: String?,
    @field:ColumnInfo
    var measurements: String?,
    @field:ColumnInfo
    var measurementUnit: String?,
    @field:ColumnInfo
    var date: String?) {

```

Таблица Record:

```

@Entity
data class RecordsTable(@field:PrimaryKey
    var recordId: Int,
    @field:ColumnInfo
    var recordValue: Double,
    @field:ColumnInfo
    var recordType: String,
    @field:ColumnInfo
    var dateOfRecord: Long,
    @field:ColumnInfo
    var workoutParentId: Int,
    @field:ColumnInfo
    var valueParentId: Int,
    @field:ColumnInfo
    var exerciseParentId: Int)

```

Таблица WorkoutSet:

```

@Entity
data class SetsTable(@field:PrimaryKey
    var workoutSetId: Int,
    @field:ColumnInfo
    var exerciseName: String,
    @field:ColumnInfo
    var workoutDataType: String,
    @field:ColumnInfo
    var dateOfCreation: Long,
    @field:ColumnInfo
    var workoutParentId: Int,
    @field:ColumnInfo
    var position: Int,
    @field:ColumnInfo
    var muscleGroup: String,
    @field:ColumnInfo
    var workoutSetVolume: Double)

```

Таблиця WorkoutSetData:

```
@Entity
data class WorkoutSetDataTable(@field:PrimaryKey
                                val valueId: Int,
                                @field:ColumnInfo
                                val setId: Int,
                                @field:ColumnInfo
                                val position: Int,
                                @field:ColumnInfo
                                val mainValue: Double,
                                @field:ColumnInfo
                                val secondaryValue: Int,
                                @field:ColumnInfo
                                val mainRecord: Boolean,
                                @field:ColumnInfo
                                val secondaryRecord: Boolean,
                                @field:ColumnInfo
                                val timeOfCreation: Long)
```

Таблиця Workout:

```
@Entity
data class WorkoutTable(@field:PrimaryKey
                        var workoutId: Int,
                        @field:ColumnInfo
                        var isEmpty: Boolean,
                        @field:ColumnInfo
                        var dateOfCreation: String,
                        @field:ColumnInfo
                        var isCompleted: Boolean,
                        @field:ColumnInfo
                        var startTime: Long,
                        @field:ColumnInfo
                        var endTime: Long,
                        @field:ColumnInfo
                        var workoutVolume: Double)
```

Створення файлу бази даних для управління всіма таблицями та командами:

```
@Database(entities = {MeasurementItem.class, ExercisesTable.class, SetsTable.class, WorkoutTable.class,
                      WorkoutSetDataTable.class, RecordsTable.class},
          version = 42, exportSchema = false)

public abstract class GymDiaryDatabase extends RoomDatabase{

    // Data Access objects:
```

```

public abstract MeasurementItemDao measurementItemDao();
public abstract ExercisesDAO exercisesDAO();
public abstract SetsDAO setsDAO();
public abstract WorkoutDAO workoutDAO();
public abstract WorkoutSetDataDAO workoutSetDataDAO();
public abstract RecordsDAO recordsDAO();

private static final String DATABASE_NAME = "MeasurementsDatabase";

private static GymDiaryDatabase instance;
public synchronized static GymDiaryDatabase getInstance(Context context){

    if(instance == null){
        instance = buildDatabase(context);
    }
    return instance;
}

private static GymDiaryDatabase buildDatabase(final Context context) {
    return Room.databaseBuilder(context,
        GymDiaryDatabase.class,
        DATABASE_NAME)
        .addCallback(new Callback() {
            @Override
            public void onCreate(@NonNull SupportSQLiteDatabase db) {
                super.onCreate(db);
                populateDB(context);
            }
        })
        //TODO: write migration class
        .fallbackToDestructiveMigration()
        .allowMainThreadQueries()
        .build();
}

public static void populateDB(Context context){
    Log.d("POPULATION", "Database populated!");
    Executors.newSingleThreadScheduledExecutor().execute() -> {
        getInstance(context).exercisesDAO().insertAll(Objects.requireNonNull(ExercisesTable.Companion.populateData()));
        getInstance(context).measurementItemDao().insertAll(Objects.requireNonNull(MeasurementItem.Companion.populateData()));
    });
}
}

```

Дана реалізація таблиць - аналог до класичної реалізації через консоль.

Висновок:

На даній лабораторній роботі я реалізував базу даних для щоденника тренувань згідно складеної раніше схеми.