

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Electiva Dispositivos móviles
Profesor: Adriana Álvarez
Laboratorio: Saving data – DB SQLite
Requisitos previos:

1. Creación de app Android.

Referencias:

<https://developer.android.com/training/basics/data-storage/databases.html>

Realice las siguientes actividades y responda las siguientes preguntas.
Para todos los puntos importe las librerías que sean necesarias.

1. Cree la siguiente clase Agenda.

```
public class Agenda {  
    private int id;  
    private String name;  
    private int phone;  
    public Agenda()  
    {  
    }  
    public Agenda(int id,String name,int phone)  
    {  
        this.id=id;  
        this.name=name;  
        this.phone=phone;  
    }  
    public void setId(int id) { this.id = id; }  
    public void setName(String name) { this.name = name; }  
    public void setPhone(int phone) { this.phone = phone; }  
    public int getId() { return id; }  
    public int getPhone() { return phone; }  
    public String getName() { return name; }  
}
```

2. Creación SQLite Database Handler. Se necesita crear una clase para realizar las operaciones de creación, lectura, actualización y borrado. Para ello, se debe crear una clase en el paquete de la aplicación como la siguiente:

```
public class DBHandler extends SQLiteOpenHelper {

    // Database Version
    private static final int DATABASE_VERSION = 1;
    // Database Name
    private static final String DATABASE_NAME = "agendaInfo";
    // Contacts table name
    private static final String TABLE_AGENDA = "agenda";
    // Shops Table Columns names
    private static final String KEY_ID = "id";
    private static final String KEY_NAME = "name";
    private static final String KEY_PHONE = "phone";
    public DBHandler(Context context) { super(context, DATABASE_NAME, null, DATABASE_VERSION); }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_AGENDA_TABLE = "CREATE TABLE " + TABLE_AGENDA + "("
            + KEY_ID + " INTEGER PRIMARY KEY," + KEY_NAME + " TEXT,"
            + KEY_PHONE + " INTEGER" + ")";
        db.execSQL(CREATE_AGENDA_TABLE);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // Drop older table if existed
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_AGENDA);
        // Creating tables again
        onCreate(db);
    }
}
```

```

// Adding new agenda
public void addAgenda(Agenda agenda) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(KEY_NAME, agenda.getName()); // Contact Name
    values.put(KEY_PHONE, agenda.getPhone()); // Contact Phone Number
    // Inserting Row
    db.insert(TABLE_AGENDA, null, values);
    db.close(); // Closing database connection
}

// Getting one shop
public Agenda getAgenda(int id) {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.query(TABLE_AGENDA, new String[] { KEY_ID,
        KEY_NAME, KEY_PHONE }, KEY_ID + "=?",
        new String[] { String.valueOf(id) }, null, null, null, null);
    if (cursor != null)
        cursor.moveToFirst();
    Agenda agenda = new Agenda(Integer.parseInt(cursor.getString(0)),
        cursor.getString(1), Integer.parseInt(cursor.getString(2)));
    / return shop
    return agenda;
}

```

```

    public List<Agenda> getAllShops() {
        List<Agenda> shopList = new ArrayList<>();
        // Select All Query
        String selectQuery = "SELECT * FROM " + TABLE_AGENDA;
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery(selectQuery, null);
        // looping through all rows and adding to list
        if (cursor.moveToFirst()) {
            do {
                Agenda agenda = new Agenda();
                agenda.setId(Integer.parseInt(cursor.getString(0)));
                agenda.setName(cursor.getString(1));
                agenda.setPhone(Integer.parseInt(cursor.getString(2)));
            } while (cursor.moveToNext());
        }
        // Adding contact to list
        shopList.add(agenda);
    }
    // return contact list
    return shopList;
}

// Getting shops Count
public int getShopsCount() {
    String countQuery = "SELECT * FROM " + TABLE_AGENDA;
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(countQuery, null);
    cursor.close();
    // return count
    return cursor.getCount();
}

// Updating a agenda
public int updateShop(Agenda agenda) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(KEY_NAME, agenda.getName());
    values.put(KEY_PHONE, agenda.getPhone());
    // updating row
    return db.update(TABLE_AGENDA, values, KEY_ID + " = ?",
        new String[] {String.valueOf(agenda.getId())});
}

// Deleting a shop
public void deleteShop(Agenda agenda) {
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(TABLE_AGENDA, KEY_ID + " = ?",
        new String[] {String.valueOf(agenda.getId())});
    db.close();
}

```

3. Programe el siguiente código dentro del onCreate del mainActivity. Importe las librerías que sean necesarias y cree las variables requeridas dentro de la clase MainActivity.

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        DBHelper db = new DBHelper(this);

        // Inserting Contacts/Rows
        Log.d("Insert: ", "Inserting ..");
        db.addAgenda(new Agenda(1, "Denver", 22345667));
        db.addAgenda(new Agenda(2, "Milán", 86774444));
        db.addAgenda(new Agenda(3, "Martin", 66666666));
        db.addAgenda(new Agenda(4, "Roma", 566667733));
        db.addAgenda(new Agenda(5, "Pepe", 94444442));
        db.addAgenda(new Agenda(6, "Vito", 99440404));

        // Reading all contacts
        Log.d("Reading: ", "Reading all contacts...");
        List<Agenda> agenda = db.getAllShops();

        for (Agenda i : agenda) {
            String log = "Id: " + i.getId() + " ,Name: " + i.getName() + " ,Phone: " + i.getPhone();
        }
        // Writing contacts to log
        Log.d("Agenda: : ", log);
    }
}
```

4. Revise la información que se despliega desde el main en **View>Tools Windows>Android** en el **logcat** tab.
5. Agregue 4 campos más a la tabla, inserte datos y lealos para imprimirlos desde el mainActivity.