Dobavljači sadržaja

Fakultet tehničkih nauka, Novi Sad

Pregled sadržaja

1 Deljena podešavanja

2 Datoteke

3 SQLite

SQLite

- Android aplikacije mogu da koriste ugrađen sistem za upravljanje bazama podataka (SQLite)
- Za razliku od većine sistema za upravljanje bazama podataka, SQLite se izvršava u istom procesu kao i aplikacija koja koristi njegove usluge
- Obezbeđuje referencijalni integritet i omogućava rad u transakcijama

sqlite3

Android SDK sadrži `sqlite3` alat koji služi za izvršavanje (SQL) naredbi:

Naredba	Opis
`.databases`	Lists names and files of attached databases.
`.tables ?TABLE?`	Lists names of tables (if TABLE is specified, only dumps tables matching LIKE pattern TABLE).
`.dump ?TABLE? ...`	Dumps the database in an SQL text format (if TABLE is specified, only dumps tables matching LIKE pattern TABLE).
`.schema ?TABLE?`	Shows the CREATE statements (if TABLE is specified, only dumps tables matching LIKE pattern TABLE).
`.backup ?DB? FILE`	Backups database (default "main") to FILE.
`.restore ?DB? FILE`	Restores content of the database (default "main") from FILE.

Tabela: Sqlite3 naredbe.

sqlite3

Android SDK sadrži `sqlite3` alat koji služi za izvršavanje (SQL) naredbi:

Naredba	Opis
`.read FILENAME`	Executes SQL in FILENAME.
`.import FILE TABLE`	Imports data from FILE into TABLE.
`.headers on|off`	Turns display of headers on or off.
`.mode MODE ?TABLE?`	Set output mode where MODE is one of: csv (comma-separated values), column (left-aligned columns), html (HTML <table> code), insert (SQL insert statements for TABLE), line (one value per line), list (values delimited by .separator string), tabs (tab-separated values) or tcl (TCL list elements)
`.nullvalue STRING`	Use STRING in place of NULL values.
`<sql statement>`	Može se izvršiti i proizvoljna SQL naredba.

Tabela: Sqlite3 naredbe.

sqlite3

```
> adb -s device_name shell
> sqlite3 /path_to_database/db_name.db
SQLite version 3.3.12
Enter ".help" for instructions
.... enter commands, then quit...
sqlite>.exit
> _
```

SQLite

- Za pravljenje, izmenu i otvaranje baze podataka koristi se `SQLiteOpenHelper` klasa
- Potrebno je implementirati neke od sledećih metoda:
  - `void onCreate(SQLiteDatabase database)`
  - `void onOpen(SQLiteDatabase database)`
  - `void onUpgrade(SQLiteDatabase database, int old_ver, int new_ver)`
  - `void onDowngrade(SQLiteDatabase database, int old_ver, int new_ver)`

SQLiteOpenHelper.java

```
1  public class ExampleOpenHelper extends SQLiteOpenHelper {
2
3    private static final String CREATE_DATABASE =
4      "create table NOTES ( " +
5      "  _id integer primary key autoincrement, " +
6      "  naslov text not null, " +
7      "  vreme  text not null, " +
8      "  tekst  text not null);";
9
10   public ExampleOpenHelper(Context context) {
11     super(context, DATABASE_NAME, null, DATABASE_VERSION);
12   }
13
14   @Override
15   public void onCreate(SQLiteDatabase db) {
16     db.execSQL(DATABASE_CREATE);
17   }
18
19   @Override
20   public void onUpgrade(SQLiteDatabase db, int old, int new) {
21     db.execSQL("DROP TABLE IF EXISTS " + NotesDbManager.DATABASE_TABLE);
22     onCreate(db);
23   }
24 }
```

SQLite

- Baza podataka predstavljena je klasom `SQLiteDatabase`.
- CRUD operacije nad bazom podataka izvršavaju se pozivom insert, query, update i delete metoda
  - `long insert(String table, String null_hack, ContentValues entry)`
  - `Cursor query(String table, String[] columns, String whereClause, String[] whereArgs, String groupBy, String having, String orderBy, String limit)`
  - `int update(String table, ContentValues values, String whereClause, String[] whereArgs)`
  - `int delete(String table, String whereClause, String[] whereArgs)`

SQLiteDatabase

```
1  // Connects to the database in write mode
2  SQLiteOpenHelper helper = new ExampleOpenHelper(this.context);
3  SQLiteDatabase db = helper.getWritableDatabase();
```

SQLiteDatabase

```
1  // Demonstrates the usage of instert method
2  ContentValues entry = new ContentValues();
3  entry.put("naslov", "Namirnice");
4  entry.put("vreme", "00:53");
5  entry.put("tekst", "Kupiti hleb i mleko.");
6  long id = db.insert(DATABASE_TABLE, null, entry);
```

SQLiteDatabase

```
1  // Demonstrates the usage of query method
2  Cursor c = db.query(
3    DATABASE_TABLE,
4    new String[] {_ID, TITLE, TIMESTAMP, TEXT},
5    "_ID = ?",
6    {id},
7    groupBy,
8    having,
9    orderBy,
10   limit);
```

SQLiteDatabase

```
1  // Demonstrates the usage of update method
2  ContentValues entry = new ContentValues();
3  entry.put("naslov", "Namirnice");
4  entry.put("vreme", "00:53");
5  entry.put("tekst", "Kupiti hleb i mleko.");
6  long id = db.update(DATABASE_TABLE, entry, whereClause, whereArgs);
```

SQLiteDatabase

```
1  // Demonstrates the usage of delete method
2  long id = db.delete(
3    DATABASE_TABLE,
4    "_ID = ?",
5    {id});
```

Kursori

- Relacija koja je rezultat SQL upita predstavljena je kursorom (`Cursor`)
- Kursori se koriste za navigaciju kroz rezultat upita:
  - `boolean move(int offset)`
  - `boolean moveToFirst()`
  - `boolean moveToLast()`
  - `boolean moveToNext()`
  - `boolean moveToPrevious()`
- kao i za čitanje rezultata upita:
  - `int getCount()`
  - `int getColumnIndex(String column_name)`
  - `String getColumnName(int column_index)`
  - `String getString(int column_index)`
  - `int getInt(int column_index)`
  - `long getLong(int column_index)`
  - `float getFloat(int column_index)`
  - `double getDouble(int column_index)`