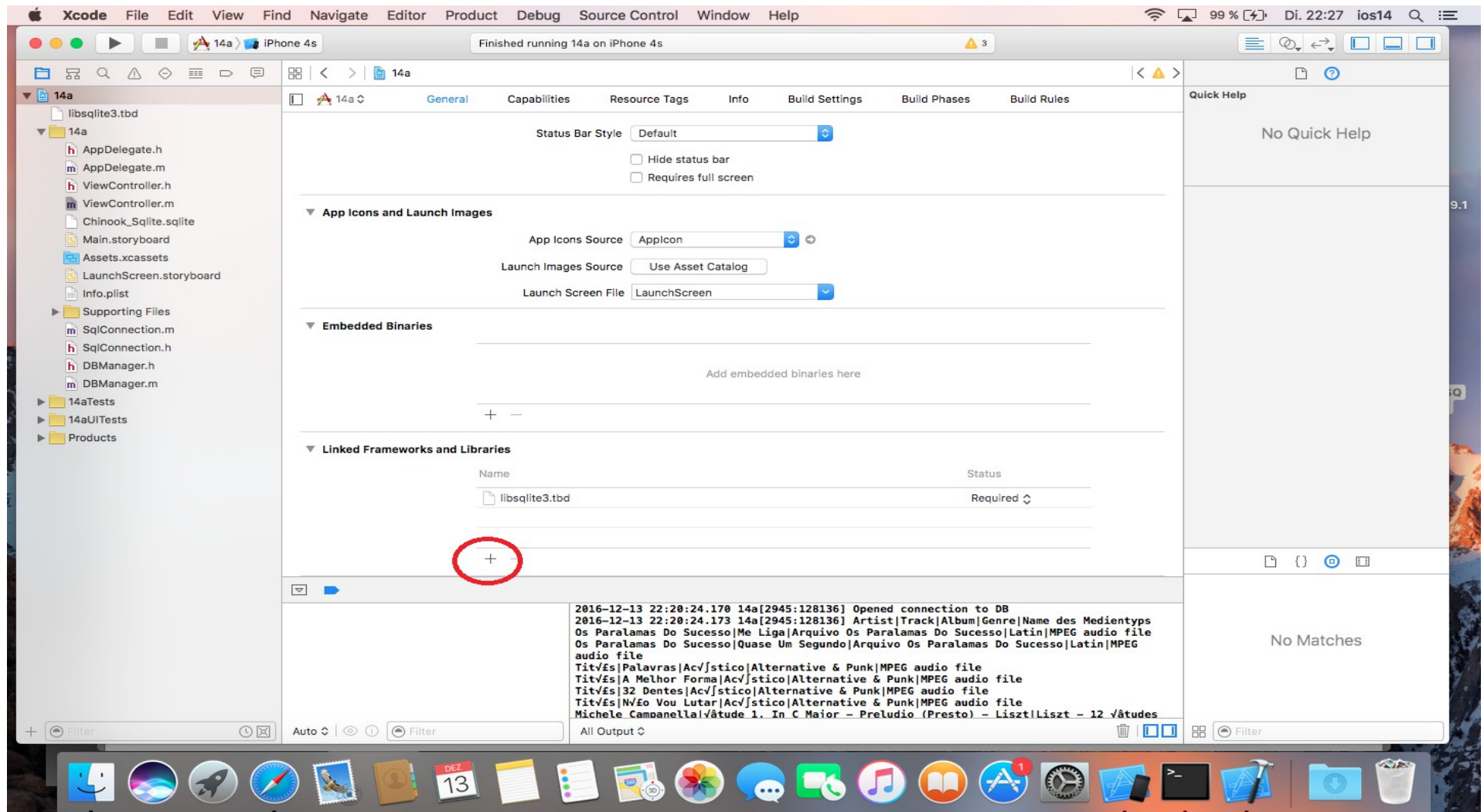


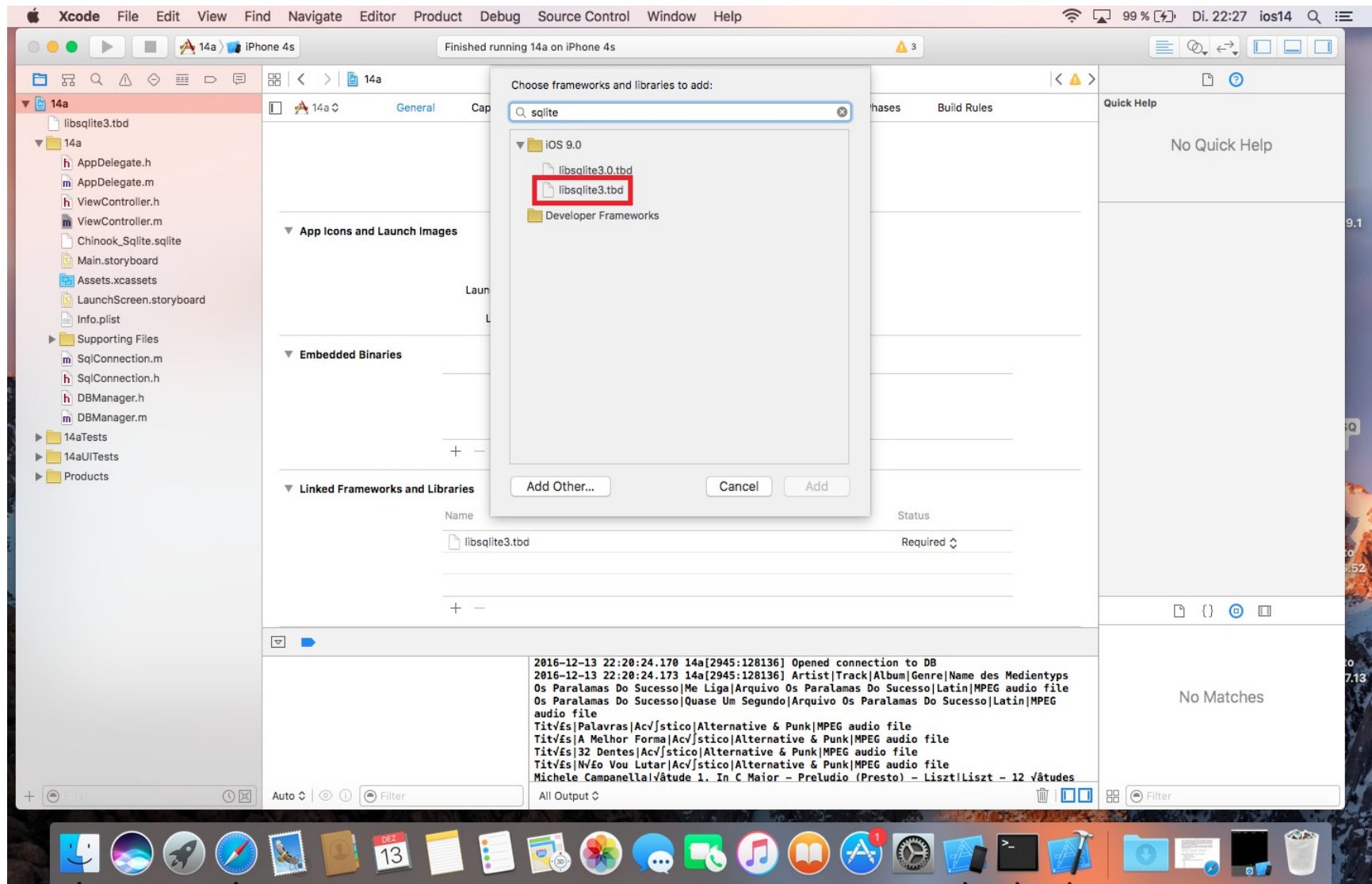
Aufgabe 14a

Sqlite mit Objective C

Sqlite Framework einbinden



Sqlite Framework einbinden



Aufbau einer DB Verbindung

```
#import <sqlite3.h>

...

// Path to Database File
NSString *databasePath = @"/Users/ios14/Desktop/ChinookDatabase1/Chinook_Sqlite.sqlite";

// Reference to DB
sqlite3 *db;

// check if DB File exists
NSFileManager *filemgr = [NSFileManager defaultManager];

if ([filemgr fileExistsAtPath:databasePath] == NO) {
    NSLog(@"DB not found");
    return;
}

// open a database connection
if (sqlite3_open([databasePath UTF8String], &db) == SQLITE_OK) {
    NSLog(@"Opened connection to DB");

    NSString *query_stmt = [NSString stringWithFormat:@"SELECT ArtistId, Name FROM Artist"];

    sqlite3_stmt *statement;

    // Convert NSString to a SQLite Statement
    if (sqlite3_prepare_v2(db, [query_stmt UTF8String], -1, &statement, nil) == SQLITE_OK) {
        // process query...

        // remove statement from memory
        sqlite3_finalize(statement);
    }
} else {
    NSLog(@"Could not open connection to DB");
}

sqlite3_close(db);
```

Die Datenbank

The screenshot shows the DB Browser for SQLite interface. The main window displays the database structure for 'Chinook_Sqlite.sqlite'. The left pane shows a tree view of tables and indexes. The right pane shows a cell editor for a selected cell, with a dropdown menu set to 'Text' and buttons for 'Importieren', 'Exportieren', and 'Auf NULL setzen'. Below the cell editor is a 'DB Schema' pane showing a list of tables and their schemas.

DB Browser for SQLite - /Users/ios14/Desktop/ChinookDatabase1/Chinook_Sqlite.sqlite

Neue Datenbank | Datenbank öffnen | Änderungen schreiben | Änderungen rückgängig machen

Datenbankstruktur | Daten durchsuchen | Pragmas bearbeiten | SQL ausführen

Tabelle erstellen | Index erstellen | Tabelle ändern

Name	Typ	Schema
Tabellen (11)		
Album		CREATE TAB
AlbumId	INTEGER	'AlbumId' INT
Title	NVARCHAR(...	'Title' NVARC
ArtistId	INTEGER	'ArtistId' INTE
Artist		CREATE TAB
ArtistId	INTEGER	'ArtistId' INTE
Name	NVARCHAR(...	'Name' NVAF
Customer		CREATE TAB
Employee		CREATE TAB
Genre		CREATE TAB
Invoice		CREATE TAB
InvoiceLine		CREATE TAB
MediaType		CREATE TAB
Playlist		CREATE TAB
PlaylistTrack		CREATE TAB
Track		CREATE TAB
Indizes (22)		
IFK_AlbumArtistId		CREATE IND
IFK_CustomerSupportRepId		CREATE IND
IFK_EmployeeReportsTo		CREATE IND
IFK_InvoiceCustomerId		CREATE IND

Datenbankzelle bearbeiten

Modus: Text | Importieren | Exportieren | Auf NULL setzen

Art der Daten in dieser Zelle: NULL
0 Bytes

Übernehmen

DB Schema

Name	Typ	Schema
Tabellen (11)		
Album		CREATE TA..
Artist		CREATE TA..
Customer		CREATE TA..
Employee		CREATE TA..
Genre		CREATE TA..
Invoice		CREATE TA..
InvoiceLine		CREATE TA..
MediaType		CREATE TA..
Playlist		CREATE TA..

SQL-Log | Diagramm | DB Schema

UTF-8

Die Datenbank

The screenshot shows the DB Browser for SQLite interface. The main window displays the 'Customer' table with 11 rows of data. The table has columns: CustomerId, FirstName, LastName, Company, and Address. The data is as follows:

CustomerId	FirstName	LastName	Company	Address
1	Luis	Gonçalves	Embraer - Empresa Brasileira de Aero...	Av. Brigadeiro Lima, 2170
2	Leonie	Köhler	NULL	Theodor-Heu Straße 34
3	François	Tremblay	NULL	1498 rue Bél
4	Bjørn	Hansen	NULL	Ullevålsveien
5	František	Wichterlová	JetBrains s.r.o.	Klanova 9/50
6	Helena	Holý	NULL	Rilská 3174/E
7	Astrid	Gruber	NULL	Rotenturmstr 1010 Innere S
8	Daan	Peeters	NULL	Grétrystraat 6
9	Kara	Nielsen	NULL	Sønder Boule 51
10	Eduardo	Martins	Woodstock Discos	Rua Dr. Falcá Filho, 155
11	Alexandre	Rocha	Banco do Brasil S.A.	Av. Paulista, 1

The interface also shows a 'Datenbankzelle bearbeiten' window with 'Modus: Text' and 'Auf NULL setzen' button. Below it, the 'DB Schema' window shows a list of tables: Album, Artist, Customer, Employee, Genre, Invoice, InvoiceLine, MediaType, and Playlist, each with a 'CREATE TA...' button. The bottom right corner shows 'UTF-8' encoding.

Effiziente SQL-Statements

```
NSString *query_michelle = [NSString stringWithFormat:@"SELECT Artist.Name as Artist, Track.Name as Track, Album.Title as Album, Genre.Name as Genre, MediaType.Name as 'Name des Medientyps' FROM Album, Artist, Track, Genre, MediaType, Customer, Invoice, InvoiceLine WHERE Album.ArtistId=Artist.ArtistId AND Track.AlbumId=Album.AlbumId AND Track.GenreId=Genre.GenreId AND Track.MediaTypeId=MediaType.MediaTypeId AND Customer.CustomerId=Invoice.CustomerId AND Invoice.InvoiceId=InvoiceLine.InvoiceId AND Track.TrackId=InvoiceLine.TrackId AND Customer.FirstName='Michelle' AND Customer.LastName='Brooks' AND Customer.City='New York'"];
```

```
NSString *query_fynn = [NSString stringWithFormat:@"SELECT Artist.Name as Artist, Track.Name as Track, Album.Title as Album, Genre.Name as Genre, MediaType.Name as 'Name des Medientyps' FROM Album, Artist, Track, Genre, MediaType, Customer, Invoice, InvoiceLine WHERE Album.ArtistId=Artist.ArtistId AND Track.AlbumId=Album.AlbumId AND Track.GenreId=Genre.GenreId AND Track.MediaTypeId=MediaType.MediaTypeId AND Customer.CustomerId=Invoice.CustomerId AND Invoice.InvoiceId=InvoiceLine.InvoiceId AND Track.TrackId=InvoiceLine.TrackId AND Customer.FirstName='Fynn' AND Customer.LastName='Zimmermann' AND Customer.City='Frankfurt'"];
```

```
NSString *query_lucas = [NSString stringWithFormat:@"SELECT Artist.Name as Artist, Track.Name as Track, Album.Title as Album, Genre.Name as Genre, MediaType.Name as 'Name des Medientyps' FROM Album, Artist, Track, Genre, MediaType, Customer, Invoice, InvoiceLine WHERE Album.ArtistId=Artist.ArtistId AND Track.AlbumId=Album.AlbumId AND Track.GenreId=Genre.GenreId AND Track.MediaTypeId=MediaType.MediaTypeId AND Customer.CustomerId=Invoice.CustomerId AND Invoice.InvoiceId=InvoiceLine.InvoiceId AND Track.TrackId=InvoiceLine.TrackId AND Customer.FirstName='Lucas' AND Customer.LastName='Mancini' AND Customer.City='Rome'"];
```

Datenbankanfrage und Konsolenausgabe

```
sqlite3_stmt *statement;
NSString *output = @"";

// Convert NSString to a SQLite Statement
if (sqlite3_prepare_v2(db, [query_lucas UTF8String], -1, &statement, nil) == SQLITE_OK) {
    int i = 0;

    output = [NSString stringWithFormat:@"%s%s%s%s%s", sqlite3_column_name(statement, 0), sqlite3_column_name
        (statement, 1), sqlite3_column_name(statement, 2), sqlite3_column_name(statement, 3), sqlite3_column_name
        (statement, 4)];

    while (sqlite3_step(statement) == SQLITE_ROW) {
        output = [NSString stringWithFormat:@"%@\n%s%s%s%s%s", output, sqlite3_column_text(statement, 0),
            sqlite3_column_text(statement, 1), sqlite3_column_text(statement, 2), sqlite3_column_text(statement, 3),
            sqlite3_column_text(statement, 4)];
        i++;
    }

    NSLog(@"%@", output);
    NSLog(@"%d entries", i);

    // remove statement from memory
    sqlite3_finalize(statement);
}
```


Ausgabe: Michelle

The screenshot shows the Xcode IDE with a Swift file named `ViewController.m` open. The code defines a method `-viewDidLoad` that uses `sqlite3` to query a database for songs. The output in the console window shows a list of 38 entries, each with a timestamp, a song name, and its genre.

```
sqlite3_stmt *statement;
NSString *output = @"";

// Convert NSString to a SQLite Statement
if (sqlite3_prepare_v2(db, [query_michelle UTF8String], -1, &statement, nil) == SQLITE_OK) {
    int i = 0;

    output = [NSString stringWithFormat:@"%s%s%s%s", sqlite3_column_name(statement, 0), sqlite3_column_name(statement,
1), sqlite3_column_name(statement, 2), sqlite3_column_name(statement, 3), sqlite3_column_name(statement, 4)];
}
```

2016-12-13 22:33:03.567 14a[3024:133995] Opened connection to DB
2016-12-13 22:33:03.571 14a[3024:133995] Artist|Track|Album|Genre|Name des Medientyps
Caetano Veloso|Terra|Prenda Minha|Latin|MPEG audio file
Caetano Veloso|Eclipse Oculito|Prenda Minha|Latin|MPEG audio file
Eric Clapton|Hey Hey|Unplugged|Blues|MPEG audio file
Eric Clapton|Lonely Stranger|Unplugged|Blues|MPEG audio file
Eric Clapton|Layla|Unplugged|Blues|MPEG audio file
Eric Clapton|Walkin' Blues|Unplugged|Blues|MPEG audio file
Led Zeppelin|When The Levee Breaks|IV|Rock|MPEG audio file
Led Zeppelin|Dazed and Confused|Led Zeppelin I|Rock|MPEG audio file
Led Zeppelin|I Can't Quit You Baby|Led Zeppelin I|Rock|MPEG audio file
Led Zeppelin|The Lemon Song|Led Zeppelin II|Rock|MPEG audio file
Led Zeppelin|Ramble On|Led Zeppelin II|Rock|MPEG audio file
Led Zeppelin|Friends|Led Zeppelin III|Rock|MPEG audio file
Amy Winehouse|Rehab|Back to Black|R&B/Soul|Protected AAC audio file
Antônio Carlos Jobim|Linha Do Horizonte|Chill: Brazil (Disc 2)|Latin|MPEG audio file
Antônio Carlos Jobim|Abrir A Porta|Chill: Brazil (Disc 2)|Latin|MPEG audio file
Cássia Eller|Palavras Ao Vento|Cássia Eller - Coleção Sem Limite [Disc 2]|Latin|MPEG audio file
Cássia Eller|Metróv. Linha 743|Cássia Eller - Sem Limite [Disc 1]|Latin|MPEG audio file
Cássia Eller|Mis Penas Lloraba Yo (Ao Vivo) Soy Gitano (Tangos)|Cássia Eller - Sem Limite [Disc 1]|Latin|MPEG audio file
Deep Purple|You Keep On Moving|Come Taste The Band|Rock|MPEG audio file
Deep Purple|No No No|Fireball|Rock|MPEG audio file
Deep Purple|Nobody's Home|Knocking at Your Back Door: The Best Of Deep Purple in the 80's|Rock|MPEG audio file
Deep Purple|Maybe I'm A Leo|Machine Head|Rock|MPEG audio file
Deep Purple|Sometimes I Feel Like Screaming|Purpendicular|Rock|MPEG audio file
Deep Purple|King Of Dreams|Slaves And Masters|Rock|MPEG audio file
Deep Purple|Stormbringer|Stormbringer|Rock|MPEG audio file
Deep Purple|The Battle Rages On|The Battle Rages On|Rock|MPEG audio file
Deep Purple|One Man's Meat|The Battle Rages On|Rock|MPEG audio file
Def Leppard|When Love & Hate Collide|Vault: Def Leppard's Greatest Hits|Rock|MPEG audio file
Dennis Chambers|Otay|Outbreak|Jazz|MPEG audio file
Spyro Gyra|Rasul|Morning Dance|Jazz|MPEG audio file
Spyro Gyra|End Of Romanticism|Morning Dance|Jazz|MPEG audio file
Stevie Ray Vaughan & Double Trouble|Travis Walk|In Step|Blues|MPEG audio file
Stone Temple Pilots|Sex Type Thing|Core|Rock|MPEG audio file
Stone Temple Pilots|Piece Of Pie|Core|Rock|MPEG audio file
System Of A Down|B.Y.O.B.|Mezmerize|Metal|MPEG audio file
System Of A Down|Question!|Mezmerize|Metal|MPEG audio file
Terry Bozzio, Tony Levin & Steve Stevens|Duende|[1997] Black Light Syndrome|Rock|MPEG audio file
The Black Crowes|Sting Me|Live [Disc 1]|Blues|MPEG audio file
2016-12-13 22:33:03.582 14a[3024:133995] 38 entries

Quick Help
Declared In ViewController.m

No Matches

Ausgabe: Fynn

The screenshot shows the Xcode IDE interface. The top status bar indicates the app is running on an iPhone 4s. The left sidebar shows a project structure with a file named `ViewController.m` selected. The main editor displays the following Swift code:

```
sqlite3_stmt *statement;
NSString *output = @"";

// Convert NSString to a SQLite Statement
if (sqlite3_prepare_v2(db, [query_fynn UTF8String], -1, &statement, nil) == SQLITE_OK) {
    int i = 0;

    output = [NSString stringWithFormat:@"%s%s%s%s", sqlite3_column_name(statement, 0), sqlite3_column_name(statement,
1), sqlite3_column_name(statement, 2), sqlite3_column_name(statement, 3), sqlite3_column_name(statement, 4)];
}
```

A warning message is visible: `Unused variable 'query_lucas'`. The console output shows a list of 38 database entries, including album and track information for various artists like Chico Buarque, Creedence Clearwater Revival, and Led Zeppelin. The console text ends with: `2016-12-13 22:35:05.433 14a[3085:136417] 38 entries`. The right sidebar shows a 'Quick Help' section with the text 'Declared In ViewController.m' and a 'No Matches' message at the bottom.

Ausgabe: Lucas

The screenshot shows the Xcode IDE interface. The top status bar indicates "Finished running 14a on iPhone 4s". The left sidebar shows a project structure for "14a" with files like AppDelegate.h, ViewController.h, and ViewController.m. The main editor displays the code for ViewController.m, which includes a SQLite query and logging statements. The console window shows a log of database entries, including album titles and genres. The search panel on the right shows "No Quick Help" and "No Matches".

```
sqlite3_column_text(statement, 1), sqlite3_column_text(statement, 2), sqlite3_column_text(statement, 3),
sqlite3_column_text(statement, 4)];
    i++;
}
NSLog(@"%@", output);
NSLog(@"%d entries", i);

// remove statement from memory
sqlite3_finalize(statement);
```

2016-12-13 22:20:24.170 14a[2945:128136] Opened connection to DB
2016-12-13 22:20:24.173 14a[2945:128136] Artist|Track|Album|Genre|Name des Medientyps
Os Paralamas Do Sucesso|Me Liga|Arquivo Os Paralamas Do Sucesso|Latin|MPEG audio file
Os Paralamas Do Sucesso|Quase Um Segundo|Arquivo Os Paralamas Do Sucesso|Latin|MPEG audio file
Titãs|Palavras|Acústico|Alternative & Punk|MPEG audio file
Titãs|A Melhor Forma|Acústico|Alternative & Punk|MPEG audio file
Titãs|32 Dentes|Acústico|Alternative & Punk|MPEG audio file
Titãs|N'Éo Vou Lutar|Acústico|Alternative & Punk|MPEG audio file
Michele Campanella|Vátude 1, In C Major - Preludio (Presto) - Liszt|Liszt - 12 Vátudes D'Execution Transcendente|Classical|Purchased AAC audio file
Emerson String Quartet|String Quartet No. 12 in C Minor, D. 783 "Quartettatz": II. Andante - Allegro assai|Schubert: The Late String Quartets & String Quintet (3 CD's)|Classical|Protected AAC audio file
AC/DC|For Those About To Rock (We Salute You)|For Those About To Rock We Salute You|Rock|MPEG audio file
Accept|Princess of the Dawn|Restless and Wild|Rock|Protected AAC audio file
AC/DC|Snowballed|For Those About To Rock We Salute You|Rock|MPEG audio file
AC/DC|Night of the Long Knives|For Those About To Rock We Salute You|Rock|MPEG audio file
Metallica|Motorbreath|Kill 'Em All|Metal|MPEG audio file
Queen|Fight From The Inside|News Of The World|Rock|MPEG audio file
Queen|Sleep On The Sidewalk|News Of The World|Rock|MPEG audio file
The Clash|This Is Radio Clash|The Singles|Alternative & Punk|MPEG audio file
The Cult|Nico|Beyond Good And Evil|Rock|MPEG audio file
The Cult|Lil' Evil|Pure Cult: The Best Of The Cult (For Rockers, Ravers, Lovers & Sinners) [UK]|Rock|MPEG audio file
The Cult|Love|Pure Cult: The Best Of The Cult (For Rockers, Ravers, Lovers & Sinners) [UK]|Rock|MPEG audio file
The Doors|The Crystal Ship|The Doors|Rock|MPEG audio file
The Police|Roxanne|The Police Greatest Hits|Rock|MPEG audio file
The Police|Every Breath You Take|The Police Greatest Hits|Rock|MPEG audio file
The Rolling Stones|As Tears Go By|Hot Rocks, 1964-1971 (Disc 1)|Rock|MPEG audio file
The Rolling Stones|You Got Me Rocking|No Security|Rock|MPEG audio file
The Rolling Stones|Respectable|No Security|Rock|MPEG audio file
The Rolling Stones|Moon Is Up|Voodoo Lounge|Rock|MPEG audio file
The Rolling Stones|Mean Disposition|Voodoo Lounge|Rock|MPEG audio file
The Tea Party|Release|Tangents|Alternative & Punk|MPEG audio file
The Tea Party|Psychopomp|Transmission|Alternative & Punk|MPEG audio file
Eric Clapton|After Midnight|The Cream Of Clapton|Blues|MPEG audio file
Eric Clapton|Promises|The Cream Of Clapton|Blues|MPEG audio file
Eric Clapton|Lonely Stranger|Unplugged|Blues|MPEG audio file
Eric Clapton|San Francisco Bay Blues|Unplugged|Blues|MPEG audio file
Faith No More|Last Cup Of Sorrow|Album Of The Year|Alternative & Punk|MPEG audio file
Faith No More|Got That Feeling|Album Of The Year|Alternative & Punk|MPEG audio file
Faith No More|Midlife Crisis|Angel Dust|Alternative & Punk|MPEG audio file
Faith No More|Be Aggressive|Angel Dust|Alternative & Punk|MPEG audio file
Faith No More|Get Out|King For A Day Fool For A Lifetime|Rock|MPEG audio file
2016-12-13 22:20:24.189 14a[2945:128136] 38 entries

Quick Help
No Quick Help
Search Documentation

No Matches

All Output