

# Doing More with SQLite Queries

---



**Jim Wilson**

MOBILE SOLUTIONS DEVELOPER & ARCHITECT

@hedgehogjim [blog.jwhh.com](http://blog.jwhh.com)



# What to Expect from This Module



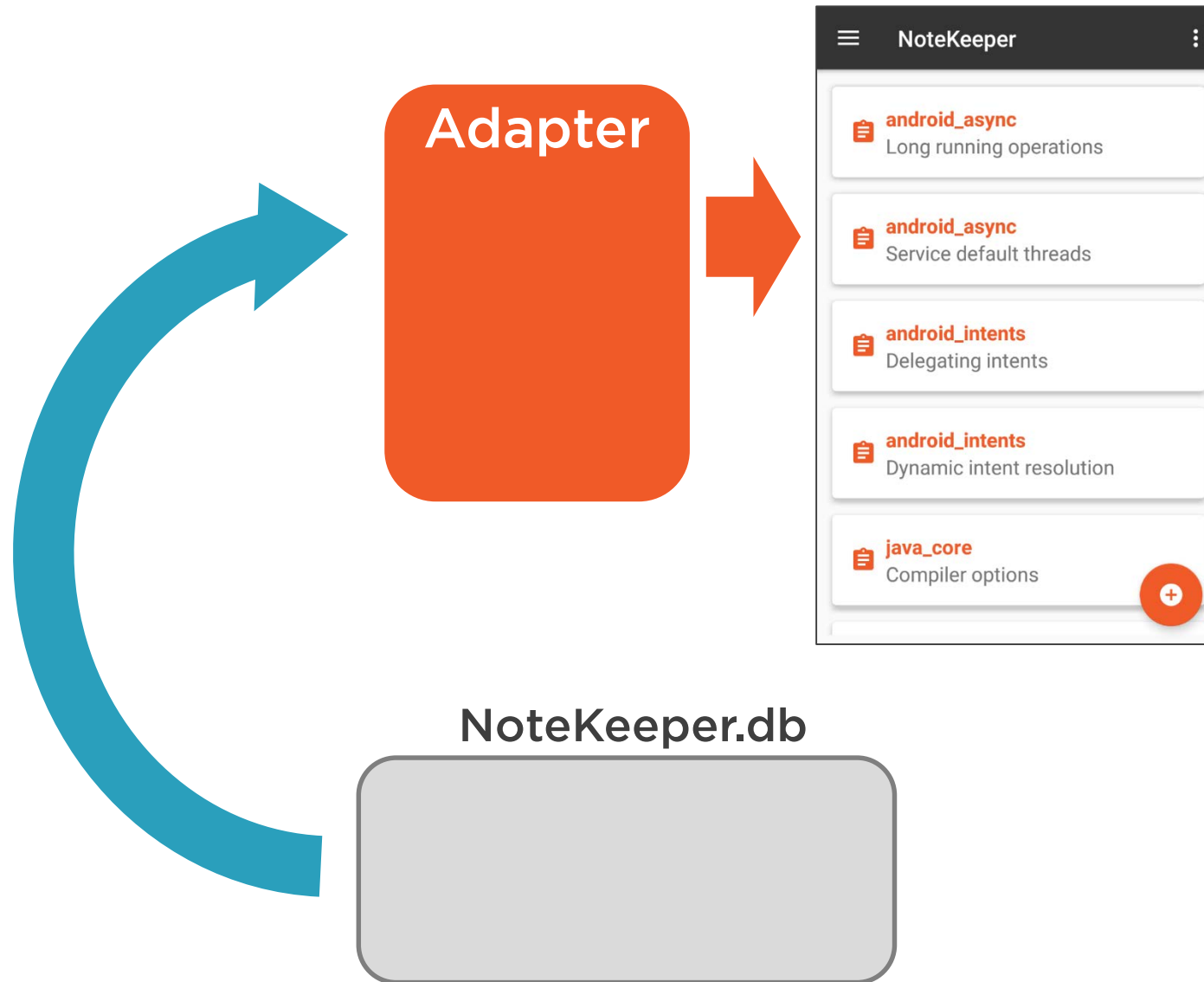
**Joining Tables**

**Column Names in Joins**

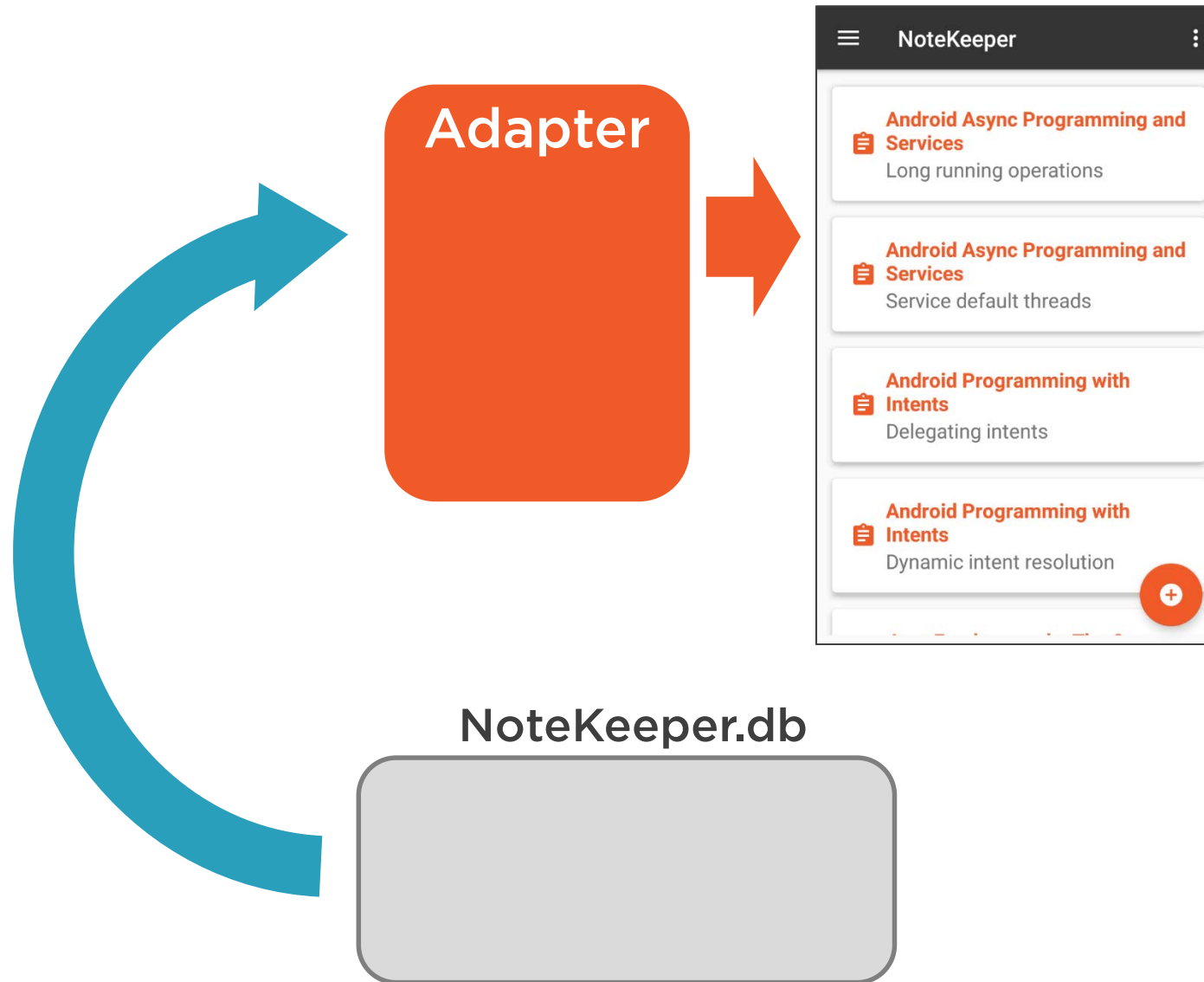
**Improving Query Performance with Indexes**

**Creating Table Indexes**

# Note in Our Main Activity



# Note in Our Main Activity



# Our Database

note\_info

<b>_id</b>	<b>note_title</b>	<b>note_text</b>	<b>course_id</b>
1	Dynamic intent resolution	Wow, intents allow components...	android_intents
2	Anonymous classes	Anonymous classes simplify...	java_lang
3	Delegating intents	PendingIntents are powerful...	android_intents



# Our Database

note\_info

_id	note_title	note_text	course_id
1	Dynamic intent resolution	Wow, intents allow components...	android_intents
2	Anonymous classes	Anonymous classes simplify...	java_lang
3	Delegating intents	PendingIntents are powerful...	android_intents

course\_info

_id	course_id	course_title
1	android_intents	Android Programming with Intents
2	android_async	Android Async Programming and Services
3	java_lang	Java Fundamentals: The Java Language
4	java_core	Java Fundamentals: The Core Platform



# Joining Tables

```
query ("note_info JOIN course_info
      ON note_info.course_id = course_info.course_id", . . .);
```



Dynamic intent resolution	android_intents	Android Programming with Intents
Anonymous classes	java_lang	Java Fundamentals: The Java Language
Delegating intents	android_intents	Android Programming with Intents

# Column Names in Joins

## Column names may be ambiguous in a join

- Tables may duplicate column names
- Need a way to disambiguate names

## Column names can be table qualified

- Precede column name with table name
- Separate with a dot (i.e. period)

`note_info.course_id`

`course_info.course_id`





# Column Names in Joins

## When to use table qualified names

- Specifying columns in join statement
- Specifying desired list of columns

## When not to use table qualified names

- Don't pass to getColumnIndex



# Our note\_info Table


```
CREATE TABLE note_info (  
    _id INTEGER PRIMARY KEY,  
    note_title TEXT NOT NULL,  
    note_text TEXT,  
    course_id TEXT NOT NULL )
```



# Our note\_info Table

Primary  
Key

note\_info

	<b>_id</b>	<b>note_title</b>	<b>note_text</b>	<b>course_id</b>
	1	Dynamic intent resolution	Wow, intents allow components...	android_intents
	2	Anonymous classes	Anonymous classes simplify...	java_lang
	3	Delegating intents	PendingIntents are powerful...	android_intents
	4	Parameters	Leverage variable-length param...	java_lang
	5	Long running operations	Foreground services can be tie...	android_async
	6	Serialization	Remember to include SerialVer...	java_core
	7	Service default threads	Did you know that by default A...	android_async
	8	Compiler options	The -jar option isn't compatible...	java_core



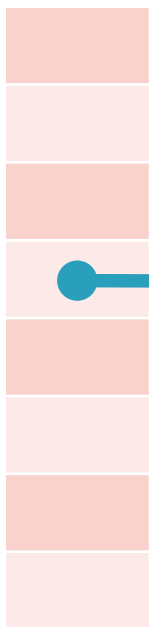
# Our note\_info Table

note_info			
_id	note_title	note_text	course_id
1	Dynamic intent resolution	Wow, intents allow components...	android_intents
2	Anonymous classes	Anonymous classes simplify...	java_lang
3	Delegating intents	PendingIntents are powerful...	android_intents
4	Parameters	Leverage variable-length param...	java_lang
5	Long running operations	Foreground services can be tie...	android_async
6	Serialization	Remember to include SerialVer...	java_core
7	Service default threads	Did you know that by default A...	android_async
8	Compiler options	The -jar option isn't compatible...	java_core



# Our note\_info Table

Index on  
note\_title



note_info				
_id	note_title	note_text	course_id	
1	Dynamic intent resolution	Wow, intents allow components...	android_intents	
2	Anonymous classes	Anonymous classes simplify...	java_lang	
3	Delegating intents	PendingIntents are powerful...	android_intents	
4	Parameters	Leverage variable-length param...	java_lang	
5	Long running operations	Foreground services can be tie...	android_async	
6	Serialization	Remember to include SerialVer...	java_core	
7	Service default threads	Did you know that by default A...	android_async	
8	Compiler options	The -jar option isn't compatible...	java_core	



# Indexes

## Helpful in a number of scenarios

- Column appears in selection criteria
- Column is part of query ordering
- Column is part of a join

## Indexes don't affect way query is written

- Database automatically chooses whether to utilize index(es)



# Indexes

**Indexes are sometimes created implicitly**

- Adding a unique column constraint

```
CREATE TABLE course_info (  
    _id INTEGER PRIMARY KEY,  
    course_id TEXT UNIQUE NOT NULL,  
    course_title TEXT NOT NULL )
```



# Indexes

## Indexes can be explicitly created

- Table can have as many as needed
- By default an index allows duplicates
- Can specify a uniqueness requirement
- Can span multiple columns

## Creating an index

- Use CREATE INDEX statement
- Provide name for index
- Identify table and one or more columns





# Summary



## Joining tables

- Allows us to return a cursor containing columns from multiple tables

## SQLiteDatabase.query

- Use join statement instead of table name
- Provide tables to be joined
- Provide columns to be matched



# Summary



## Column names may be ambiguous in joins

- Can table qualify column names

## When to use table qualified names

- Specifying columns in join statement
- Specifying desired list of columns

## When not to use table qualified names

- Don't pass to getColumnIndex



# Summary



## **Index can improve query performance**

- Provide efficient access to column value
- By default indexes allow duplicates
- Can specify uniqueness requirement
- Can span multiple columns

## **Indexes don't affect way query is written**

- Database automatically chooses whether to utilize index(es)

