Sun*

SQLite, Core Data

Outline

1. SQLite
2. Core Data

1. SQLite

1. SQL Database
2. Queries
3. Learn more

1.1 SQL Database

- Store data in tables of rows and columns (spreadsheet…)
- Field = intersection of a row and column
- Fields contain data, references to other fields, or references to other tables
- Rows are identified by unique IDs
- Column names are unique per table

1.1 SQL Database

1.1.1 Tables

WORD_LIST_TABLE		
_id	**word**	**definition**
1	"alpha"	"first letter"
2	"beta"	"second letter"
3	"alpha"	"particle"

1.1 SQL Database

1.1.2 SQLite software library

- Implements SQL database engine that is
  - ➔ self-contained (requires no other components)
  - ➔ serverless (requires no server backend)
  - ➔ zero-configuration (does not need to be configured for your application)
  - ➔ transactional (changes within a single transaction in SQLite either occur completely or not at all)

1.1 SQL Database

1.1.3 What is a transaction?

- A transaction is a sequence of operations performed as a single logical unit of work.
- A logical unit of work must have four properties
  - atomicity
  - consistency
  - isolation
  - durability

1.1 SQL Database

1.1.4 All or nothing

- All changes within a single transaction in SQLite either occur completely or not at all, even if the act of writing the change out to the disk is interrupted by
  - program crash
  - operating system crash
  - power failure

1.1 SQL Database

1.1.5 ACID

- **Atomicity**: All or no modifications are performed
- **Consistency**: When transaction has completed, all data is in a consistent state
- **Isolation**: Modifications made by concurrent transactions must be isolated from the modifications made by any other concurrent transactions
- **Durability**: After a transaction has completed, its effects are permanently in place in the system

1.2 Queries

1.2.1 SQL basic operations

- Insert rows
- Delete rows
- Update values in rows
- Retrieve rows that meet given criteria

1.2 Queries

1.2.2 SQL Query

- SELECT word, description
  FROM WORD_LIST_TABLE
  WHERE word="alpha"

Generic

- SELECT columns
  FROM table
  WHERE column="value"

1.2 Queries

1.2.3 SELECT columns FROM table

- **SELECT columns**
  - ➔ Select the columns to return
  - ➔ Use * to return all columns

- **FROM table** — specify the table from which to get results

1.2 Queries

1.2.3 SELECT columns FROM table

- **WHERE** — keyword for conditions that have to be met

- **column="value"** — the condition that has to be met
  - ➔ common operators: **=, LIKE, <, >**

1.2 Queries

1.2.4 AND, ORDER BY, LIMIT

- SELECT id FROM WORD_LIST_TABLE WHERE word="alpha" **AND** definition **LIKE** "%art%" **ORDER BY word DESC LIMIT 1**

  - **AND**, **OR** — connect multiple conditions with logic operators
  - **ORDER BY** — omit for default order, or **ASC** for ascending, **DESC** for descending
  - **LIMIT** — get a limited number of results

1.2 Queries

1.2.5 Sample queries

1	SELECT * FROM WORD_LIST_TABLE	Get the whole table
2	SELECT word, definition FROM WORD_LIST_TABLE WHERE _id > 2	Returns [["alpha", "particle"]]

1.2 Queries

1.2.5 More sample queries

3	SELECT _id FROM WORD_LIST_TABLE WHERE word="alpha" AND definition LIKE "%art%"	Return id of word alpha with substring "art" in definition [["3"]]
4	SELECT * FROM WORD_LIST_TABLE ORDER BY word DESC LIMIT 1	Sort in reverse and get first item. Sorting is by the first column (_id) [["3","alpha","particle"]]

1.2 Queries

1.2.5 Last sample query

5	SELECT * FROM WORD_LIST_TABLE LIMIT 2,1	Returns 1 item starting at position 2. Position counting starts at 1 (not zero!). Returns [["2","beta","second letter"]]

1.3 Learn more

- SQLite website
- Full description of the Query Language

2. Core data

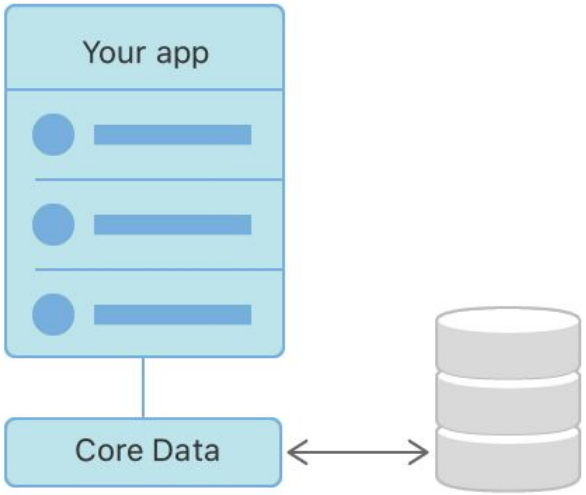
1. What is **Core Data**?
2. Advantages
3. Learn more

2.1 What is Core Data?

- A framework that you use to manage the model layer objects in you application.
- Core Data isn't the database of your application nor is it an API for persisting data to a database.
- It provides generalized and automated solutions to common tasks associated with object life cycle and object graph management, including persistence.

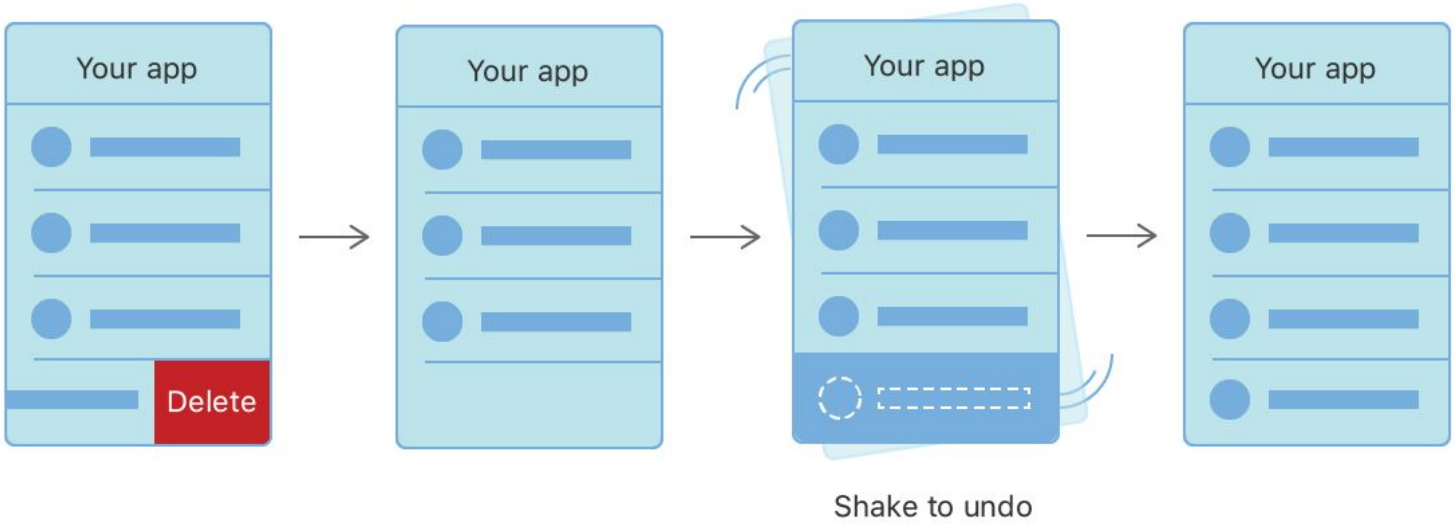
2.2 Advantages

- **Persistence**
  - ➔ Core Data abstracts the details of mapping your objects to a store, making it easy to save data from Swift and Objective-C without administering a database directly.



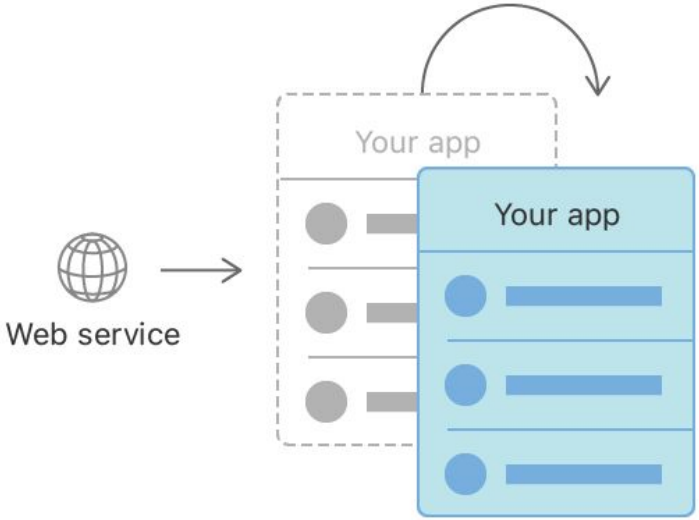
2.2 Advantages (cont)

- **Undo and Redo of Individual or Batched Changes**
  - ➔ Core Data's undo manager tracks changes and can roll them back individually, in groups, or all at once, making it easy to add undo and redo support to your app.



2.2 Advantages (cont)

- **Background Data Tasks**
  - Perform potentially UI-blocking data tasks, like parsing JSON into objects, in the background. You can then cache or store the results to reduce server roundtrips.



2.2 Advantages (cont)

- **View Synchronization**
  - ➔ Core Data also helps keep your views and data synchronized by providing data sources for table and collection views.

- **Versioning and Migration**
  - ➔ Core Data includes mechanisms for versioning your data model and migrating user data as your app evolves.

2.3 Learn more

- Apple Documentation

Question & Answer?