Persistence Strategies

1. Application data needs to be persisted on disk
2. Full in-memory object model with bulk updates
	1. Load object model from disk at startup
	2. Application features run off of in-memory object model
	3. Save object model to disk in bulk at appropriate times (e.g., user “Save”, app exit)
	4. Crash causes data loss
	5. Complete in-memory model and bulk load/save is not feasible for large data sets
	6. Only works for single-user applications
3. Full in-memory object model with incremental updates
	1. Load object model from database at startup
	2. Application features run off of in-memory object model
	3. Incremental changes to the in-memory object model are immediately saved to disk
	4. Complete in-memory model and bulk load is not feasible for large data sets
	5. Only works for single-user applications
4. Partial in-memory model with incremental updates
	1. There is no full object model in memory
	2. Application loads subsets of the object model from disk as needed to perform operations. Operations run off of a partial in-memory object model.
	3. Incremental changes to the in-memory object model are immediately saved to disk
	4. Partial in-memory object model is discarded as soon as operation is complete
	5. Scales to large data sets
	6. Works for multi-user applications
	7. Takes work to fetch the data required for each operation

Persistence Technologies

1. Serialization, XML, Custom file format, Database, Cloud storage service
2. All of these approaches are appropriate in different contexts
3. Databases are very common because they are relatively easy to use, allow incremental updates, and allow data to be shared by multiple programs in a concurrent fashion.
4. Although there are many kinds of databases, Relational Databases are the most common, so we will focus on those.

Database Management Systems

1. Databases are implemented by software packages called Database Management Systems (DBMS)
2. Commonly used relational DBMS’s include MySQL, SQL Server, and Oracle
3. DBMS’s store data in files in a way that scales to large amounts of data and allows data to be accessed efficiently
4. Embedded vs. Client/Server
5. Programmatic connection vs. Interactive management console

Relational Databases

1. Relational databases are based on the relational data model that you learned about in CS 236
2. In the object-oriented data model we have classes. Objects are instances of classes. Objects have attributes. Relationships between objects are represented as pointers.
3. In the relational data model, data is stored in tables consisting of columns and rows. Each row in a table represents an object. The columns in a row store the object’s attributes. One of an object’s attributes is its “key”, which is a unique identifier for that object. Relationships between objects are represented using keys.
4. Taken together, all the table definitions in a database make up the “schema” for the database.

SQL – Structured Query Language

1. Language for creating tables, inserting data, querying data, updating data, and deleting data

SQL CREATE TABLE

1. Data types
2. CREATE TABLE
3. Null
4. Primary keys
5. Modeling object relationships
	1. Foreign keys
	2. One-to-one
	3. One-to-many
	4. Many-to-many
6. DROP TABLE

SQL INSERT

SQL SELECT

SQL UPDATE

SQL DELETE

Programmatic Database Access

1. Connections
2. Statements
3. Result sets
4. Prepared statements

Transactions

Persistence Design Patterns

1. Data Access Object
2. Data Transfer Object