In order to communicate with a database, you've got to speak it's language. SQL is a specialized programming language used for interacting with databases for example, reading information from a database.

SQL stands for Structured Query Language. SQL has been around since the 1970s!

Reading data from a database is known as querying.

Further Reading

* [SQL Entry in Wikipedia](https://en.wikipedia.org/wiki/SQL)

SQL Database Websites

* [MySQL](http://www.mysql.com/)
* [PostgreSQL](http://www.postgresql.org/)
* [SQLite](http://www.sqlite.org/)
* [Microsoft SQL Server](https://www.microsoft.com/en-us/server-cloud/products/sql-server/)
* [Oracle](https://www.oracle.com/database/index.html)

NoSQL Database Websites

* [MongoDB](https://www.mongodb.org/)
* [CouchBase](http://www.couchbase.com/)
* [Redis](http://redis.io/)

Querying – Reading data from a db.

Many db that understand SQL

* + MySQL, PostgreSQL, MSSQL, Oracle, SQLite

Other db(NOSQL DB): more specialized methods.

Mongo DB

Couch Base

Redis

Further Reading

* [Schema](https://en.wikipedia.org/wiki/Database_schema)

How data should be stored;

Databases allow users to store data in well defined sections. This keeps the data uniform. However, the SQL you write can show the data in different ways which is handy for reporting subsets of data

Data Types:

* + Text: name, and description
  + Numeric: prices ages and quant
  + Date: time related

Types of Data

Here's a slightly larger list of data types than in the video.

* Text Type Examples
  + TEXT
  + VARCHAR
* Numeric Type Examples
  + INT
  + INTEGER
* Date Type Examples
  + DATETIME
  + DATE
  + TIMESTAMP

Here's documentation sites for some other databases where you can see the similarities and differences in data types.

* [MySQL Data Types](http://dev.mysql.com/doc/refman/5.7/en/data-types.html)
* [SQLite Data Types](http://www.sqlite.org/datatype3.html)
* [PostgreSQL Data Types](http://www.postgresql.org/docs/8.4/static/datatype.html)
* [Microsoft SQL Data Types](https://msdn.microsoft.com/en-us/library/ms187752.aspx?f=255&MSPPError=-2147217396)

Definition best suits the term db? – a computerized system for the storage and retrieval of organized information.

Different Database Tools

* [Mode Analytics](http://modeanalytics.com/) used for businesses to get insights from their database.
* [pgAdmin](http://pgadmin.org/) for PostgreSQL
* [phpMyAdmin](https://www.phpmyadmin.net/) and [MySQL Workbench](http://mysqlworkbench.org/) for MySQL

### **SQL Used**

To delete specific rows from a table:

DELETE FROM <table> WHERE <condition>;

Examples:

DELETE FROM users WHERE email = "andrew@teamtreehouse.com";

DELETE FROM movies WHERE genre = "Musical";

DELETE FROM products WHERE stock\_count = 0;

See all of the SQL used in Modifying Data With SQL in the [Modifying Data With SQL Cheatsheet](https://github.com/treehouse/cheatsheets/blob/master/modifying_data_with_sql/cheatsheet.md).

### **Definitions**

**Autocommit** - every statement you write gets saved to disk.  
**Seeding** - populating a database for the first time.  
**Script file** - a file containing SQL statements.

### **SQL Used**

Switch autocommit off and begin a transaction:

BEGIN TRANSACTION;

Or simply:

BEGIN;

To save all results of the statements after the start of the transaction to disk:

COMMIT;

See all of the SQL used in Modifying Data With SQL in the [Modifying Data With SQL Cheatsheet](https://github.com/treehouse/cheatsheets/blob/master/modifying_data_with_sql/cheatsheet.md).

#### SQL Used

To reset the state of the database to before the begining of the transaction:

ROLLBACK;

See all of the SQL used in Modifying Data With SQL in the [Modifying Data With SQL Cheatsheet](https://github.com/treehouse/cheatsheets/blob/master/modifying_data_with_sql/cheatsheet.md).

#### Definitions

**ORM** - Object-Relational Mapping – used to perform CRUD operations with a language other than SQL.  
**DML** - Data Manipulation Language - the subset of the SQL programming language that deals with CRUD operations.

#### Examples of ORMs

* [Hibernate](http://hibernate.org/) for Java
* [CoreData](https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CoreData/index.html) for Objective-C or Swift
* [Django ORM](https://docs.djangoproject.com/en/1.9/#the-model-layer) for Python
* [ActiveRecord](http://api.rubyonrails.org/classes/ActiveRecord/Base.html) for Ruby

ORM(Object-Relational Mapping)

* + Perform CRUD operations with a language other than SQL to interface with the db and perform crud

Benefits of an ORM

* + Handle transaction. Helps devs build robust applications.
  + One Language. ORM uses sql under the hood

Examples of ORMS

* + HIBERNATE : java
  + CoreData : objective C
  + Django – python
  + Active Record – ruby

DML – CRUD insert, update etc.