Amazon Web Services

Amazon RDS

Amazon Relational Database Service

- web service to easily set up, operate, manage, and scale a relational database.
- resizeable => cost-efficient
  Why?
- manages backups, software, etc.
- IAM: Identity and Access Management
- availability of different DB instance types (combos of CPU, memory, storage, and networking capabilities)
    - flexibile

Cloud Computing

- on-demand availability of computer system resources
- esp. data storage and computing power
    - with no direct management by the user

IaaS, PaaS, SaaS

Three main types of cloud computing - they provide managed, flexible, and cloud-based alternatives to having services on-premises
They are pay as you go (only pay for what you use)

Iaas: Infrastructure as a Service (*Hardware*)

- abstracts away the need for physical infrastructure.
- self-service model for managing remote data centers
    - ex. AWS
- Managed for you => Servers, Storage, Networking, and Virtualization

PaaS: Platform as a Service (*Hardware and Software)

- used mainly by application developers
- allows orgs. to build, run, and manage applications without IT infrastructure
    - ex. Google Cloud Platform (GCP) Managed for you => Runtime, OS, and everything from Iaas.

SaaS: Software as a Service (*Everything - 3rd Party Software*)

- software is licesnsed and used on a subscription basis
    - ex. Salesforce, Dropbox
- software is hosted in the cloud

- End users

Database: a collection of objects designed to hold information (like data structures)

- unlike data structures in code, a database has persistence

Relational Database: a DB that holds information in tables that are related to each other

Nonrelational Database: a DB that holds information in things collections/documents - and are unrelated. (MongoDB)

RDBMS: Relational Database Management System

- software designed to manage a database
- the way in which we interact with our database
    - Oracle SQL is an RDBMS
    - PostgreSQL (we will use this)
    - Maria DB, Amazon Aurora, MySql, Microsoft SQL Server
- There is no *core* SQL Language
- Each of these are all 90% the same language, but they have their own minor differences

SQL: Structured Query Language

- used for managing data held inside an RDMBS.
- SQL doesn't execute in its entirety (like Java)
- Instead, we have the freedom to choose how much code is executed at any given time
    - that's why we call it a query language
- SQL files more like a workspace (scripts)

Schema: a group of DB related objects - outlines to our database

Cursor: Result set of a SQL Query - what information/data is returned from a SQL query

View: a virtual table based on the result of a query (a representation of the information that you asked for)

SQL Sublanguages

SQL has 5 sublanguages. These are basically a breakdown of different keywords or commands that accomplish different tasks.

DDL: Data Definition Language

- anything that defines the rules and structures of a database (laying the foundation of our DB)
- creating, altering, or removing objects of your DB (**NOT DATA ITSELF**)

- objects => tables, sequences, procedures, etc.
- CREATE, ALTER, RENAME, DROP, TRUNCATE
- DDL statements will auto-commit

DML: Data Manipulation Language

- anything that adds, removes, or edits data in your DB
- INSERT, UPDATE, DELETE
- DML statements are not auto-committed

DQL: Data Query Language

- for reading data from DB
- SELECT

DCL: Data Control Language

- responsible for granting permission to users of your database
- not a focus for us => b/c our control is usually done through the application and not the database
- GRANT, REVOKE

TCL: Transaction Control Language

- used for generating transactions
- everytime you execute a SQL statement, you use something called a transaction
- **Transaction**: a series of commands that are performed.
- COMMIT, ROLLBACK, SAVEPOINT
- these are what allow you to finalize transactions or revert changes
- cannot roll back a commit

---

differences in SQL 'flavors' tend to be with the commands/keywords.

- Oracle SQL => DELETE ....
- PostgreSQL => DELETE FROM ...
- Datatypes can change as well

CRUD Operations (pertain to DATA and not SCHEMA)

CREATE : INSERT INTO (adding data to our table)

READ : SELECT (retrieving data from our table)

UPDATE : UPDATE ... SET (updating data in our table)

DELETE : DELETE (deleting data from our table)

```
-- DDL
CREATE TABLE myTable (
    id integer serial,
    name varchar(40),
    age integer(10)
);

--DML
INSERT INTO myTable VALUES(default, 'Dan', 195);
```

name != Name my_Table != MYTABLE

InSeRT == INSERT