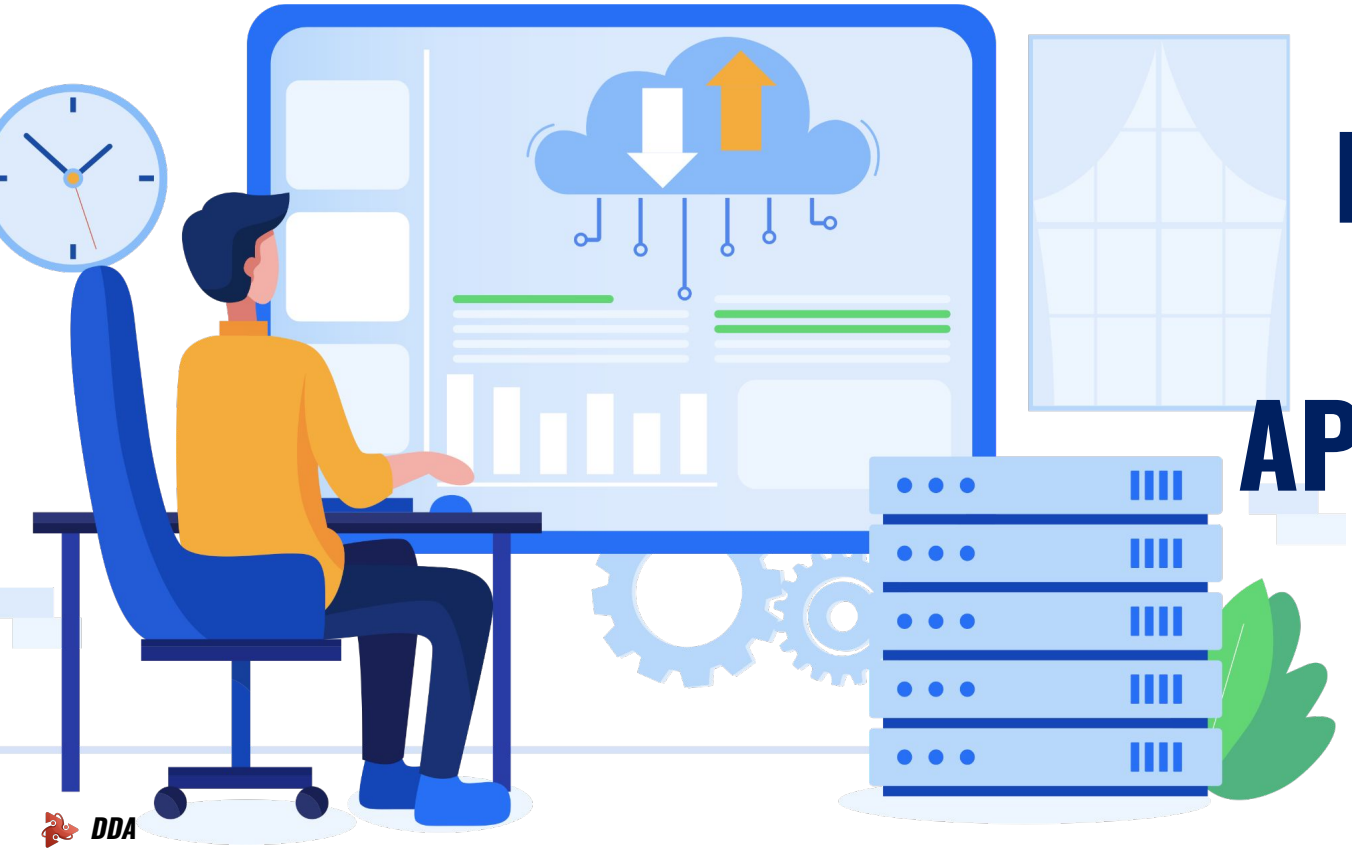
Hi!

#AMA

How R U?

#xyzabc



(DDA) 1.2

DEVELOPING DYNAMIC APPLICATIONS

2021

DDA

OVERVIEW

Key Topics aka JOB SKILLS

Firebase
Real-time Databases
Authentication
JSON
SMTP
Analytics
Firebase & Web
Full-stack Development
Unity

Learning Objectives

1. Describe and evaluate the differences and usages of SQL and NoSQL databases
2. Integrate Cloud platform into Game Engine
3. Work with authentication services for user management
4. Design appropriate data schemas for NoSQL based applications
5. Apply the concepts for designing NoSQL databases for application development
6. Transact data on Firebase
7. Perform real-time data synchronization
8. Present Firebase data in a systematic way
9. Build serverless applications using real-time databases and cloud services

DDA

SCHEDULE

Week 2 – Week 5	Assignment 1 (SOLO - 20%)
Week 5 & Week 9	Asynchronous
Week 6 – Week 10	Assignment 2 w/ ITD (GROUP - 20%)
Week 10 – 11	*Break*
Week 14 – 17	IP Assignment 3 (GROUP - 30%)
All Weeks	Continual Assessment (30%)

Cher will be on reservist from Nov 5 - Nov 17
(affected classes TBA)
***** Synchronous/F2F schedule are tentative and may subject to change***

Who?

#MODULEREP

Let's make things better

As a class, let's decide on how we want our class environment to be like. Be it virtual, or F2F. We will work towards and commit to such expectations.

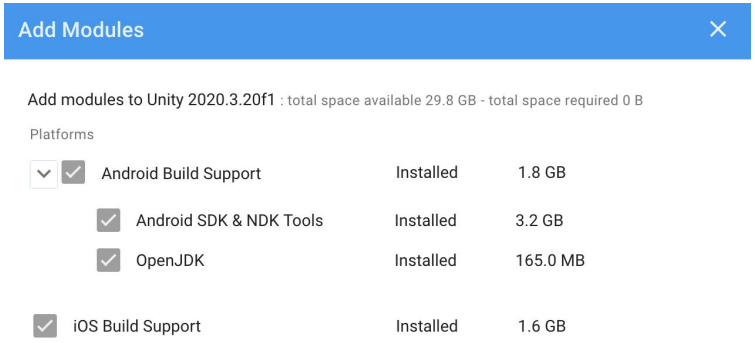
<INSERT YOUR CONTRACT TERMS>

https://bit.ly/dda-contract

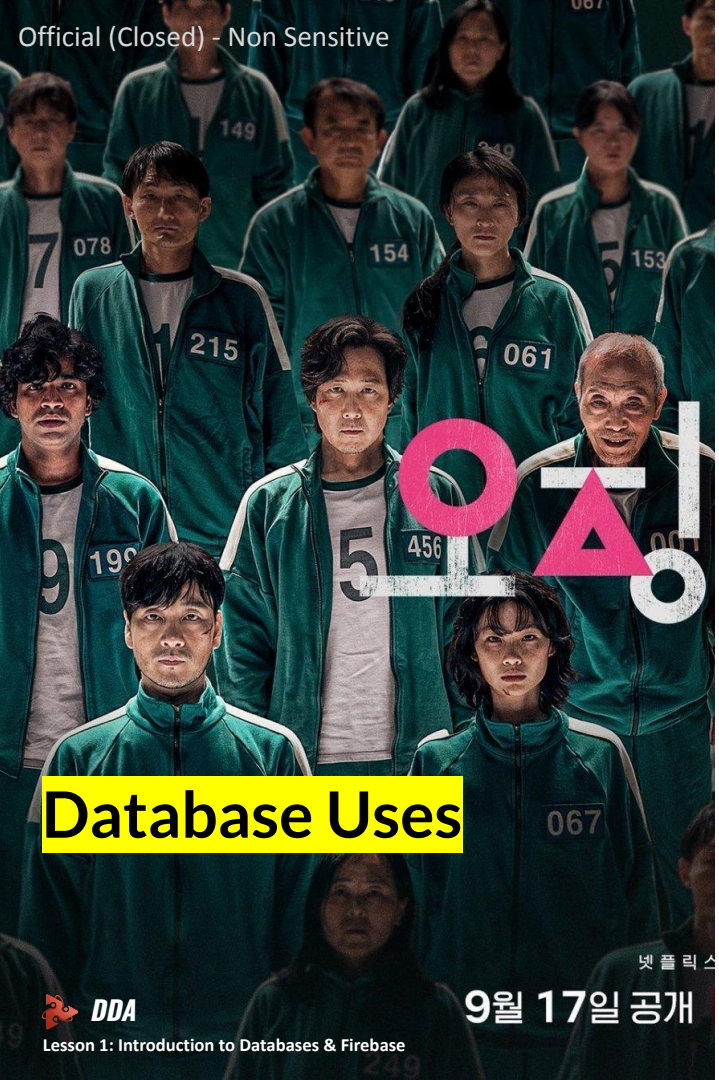
Clear your HDD
#SOFTWARE

SOFTWARE CHECK-IN

- Unity 2021.1.21 - Add in Unity Hub
- Google Account - https://accounts.google.com/SignUp
- Firebase Account - requires Google Account https://firebase.google.com/
- iOS Module for Unity - Unity Hub
- Android Module for Unity - Unity Hub
- Firebase Unity SDK - https://firebase.google.com/download/unity



So what's a #DATABASE



Database Uses

We use Database for a lot of stuff

- Storing game inventory
- Game logs or logging every “sneaky” history on your browser
- Game logs can be also known as Player Stats
- Authorisation, being aware of what account, who it is, what credentials they have
- Allow access to data to create more vibrancy

To use Databases

- A web server (Apache, Nigx)
- A cloud server (Google Cloud, Digital Ocean, AWS)
- Backend services (Firebase, Amazon Web Services, Microsoft Azure)

Why Learn Database?

- **Build** fuller fledged applications
- **Why** they are relevant
- **How** people use them

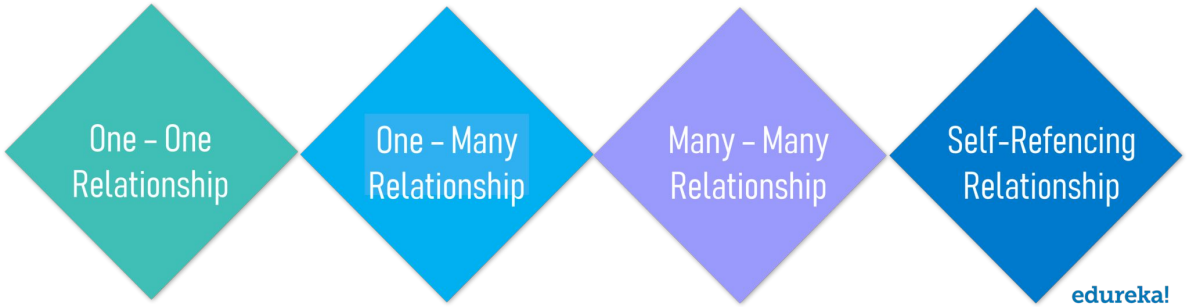
So what's #SQL

What is SQL

SQL aka Structured Query Language is the core of the relational database which is used for accessing and managing the databases.

This language is used to manipulate and retrieve data from a structured data format in the form of tables and holds relationships between those tables.

Common SQL: MYSQL, MSSQL



Reading: https://www.edureka.co/blog/sql-vs-nosql-db/

SQL Explained

What is Database & SQL https://www.youtube.com/watch?v=FR4QIeZaPeM

MySQL = A Database Management System
MySQL != SQL

There is no single database file
Each database consists of multiple files/tables

Default DBMS most hosting providers use with PHP[6]

Reference:
https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html

MySQL DataTypes

Data Type	Range	UNSIGNED
INT	-2.1 billion to 2.1 billion	0 to 4.2 billion
TINYINT	-128 to 127	0 to 255
SMALLINT	-32,768 to 32,767	0 to 65,535
MEDIUMINT	-8.3 million to 8.3 million	0 to 16.7 million

Data Type	Uses
VARCHAR	Short, variable length text
CHAR	Short, fixed length text, e.g., encrypted passwords
TEXT	Long text, e.g., articles
ENUM	One value from a predefined list
BLOB	Images, sound files, compressed files

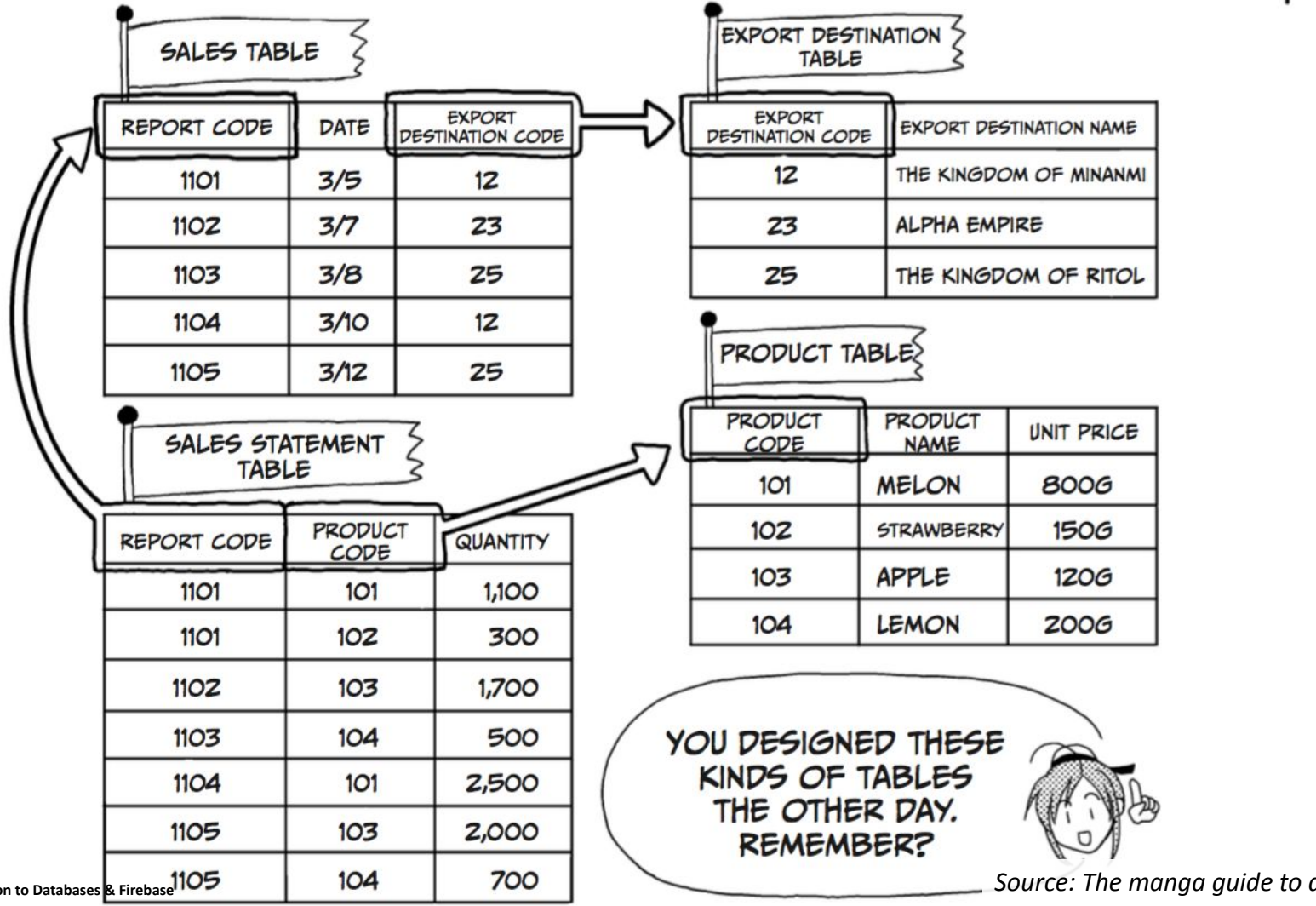
Data Type	Uses
FLOAT	Decimal values
DOUBLE	Very large decimal values
DECIMAL	Values where rounding errors are unacceptable, e.g., currency

Data Type	Uses
DATE	YYYY-MM-DD
TIME	hh:mm:ss
DATETIME	YYYY-MM-DD hh:mm:ss
TIMESTAMP	YYYY-MM-DD hh:mm:ss
YEAR	YYYY

SQL (TL;DR;)

Structured Query Language (or SQL in short) is a relational database **language** used for formulating statements that are processed by a database server

It is used by a number of commercial DBMS (Database Management System) products including Microsoft SQL Server, Oracle, SYBASE, mySQL and DB2



Source: The manga guide to databases

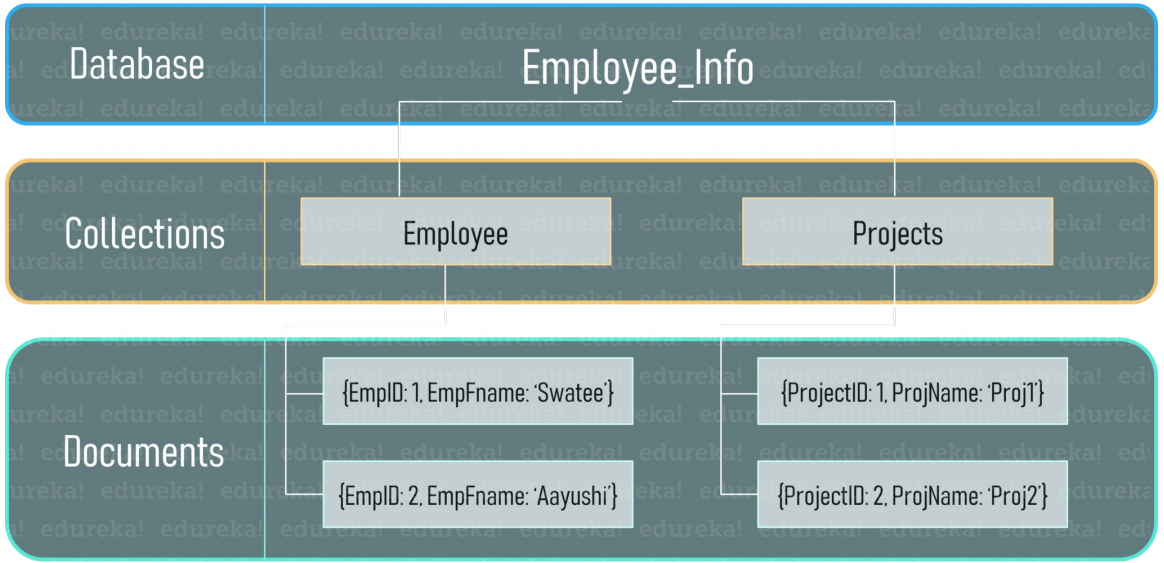
So what's #NOSQL

What is NoSQL

NoSQL, or most commonly known as Not only SQL database, provides a mechanism for storage and retrieval of unstructured data.

This type of database can handle a humongous amount of data and has a dynamic schema. So, a NoSQL database has no specific query language, no or a very few relationships, but has data stored in the format of collections and documents.

So, a database can have a 'n' number of collections and each collection can have 'm' number of documents.

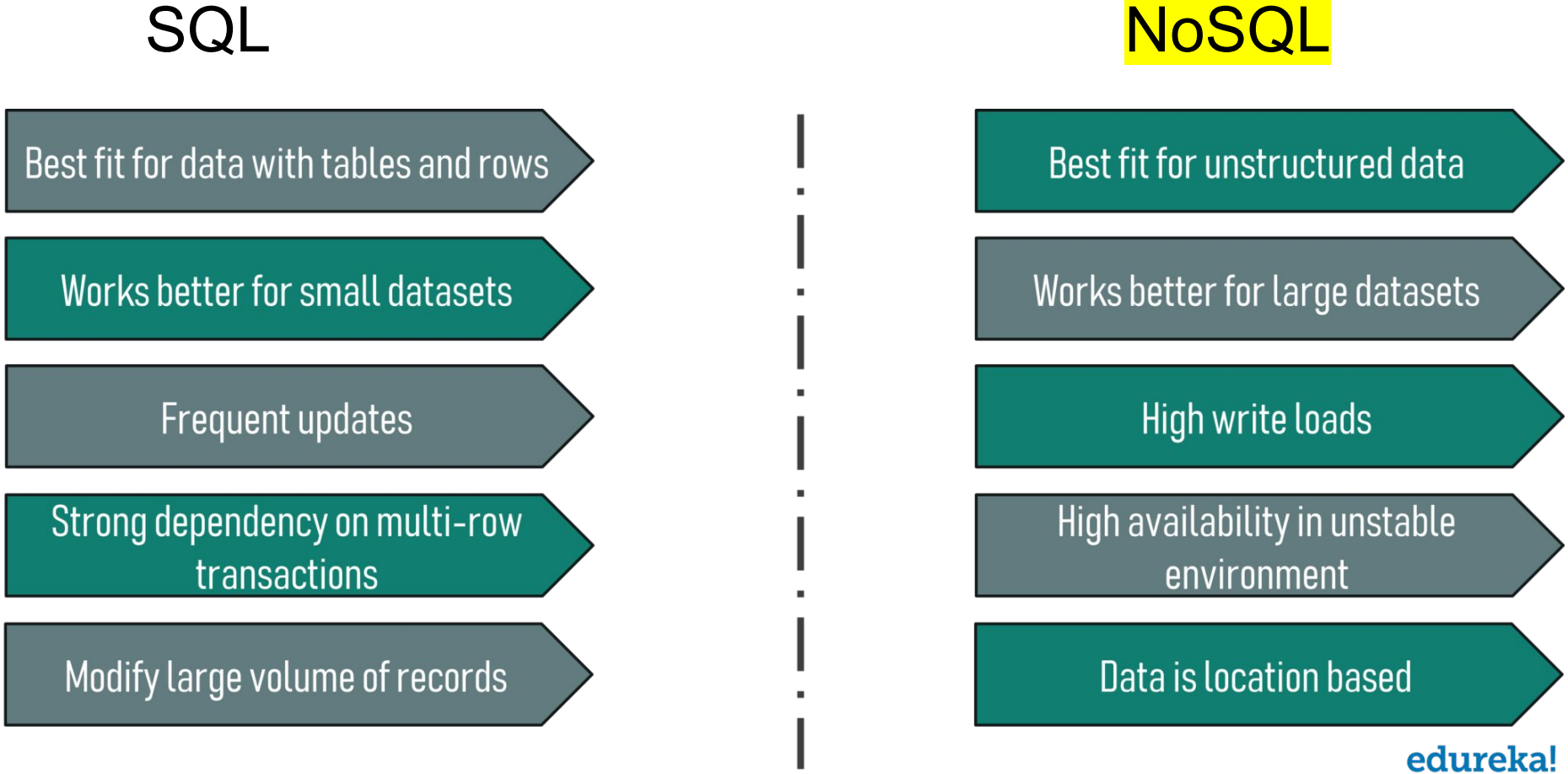


Document: JSON documents, Key-value: key-value pairs,

Reading: https://www.edureka.co/blog/sql-vs-nosql-db/

What is NoSQL

Why NoSQL DB https://youtu.be/0X43QfCfyk0


SQL
NoSQL
Best fit for data with tables and rows
Works better for small datasets
Frequent updates
Strong dependency on multi-row transactions
Modify large volume of records
Best fit for unstructured data
Works better for large datasets
High write loads
High availability in unstable environment
Data is location based
edureka!

Example of SQL

Lots of tables involved and we need to maintain relationships

order

order_id	time	amount	customer_id
1	2015-01-01 12:00:00	100.00	1
2	2015-01-02 14:00:00	200.00	1
3	2015-01-03 08:00:00	150.00	2
...	...	...	...

Your original "order" table

```
SELECT    customer_id,
          MIN(DATE(time)) AS first_order_date,
          SUM(amount) AS lifetime_amount
FROM      order
GROUP BY customer_id
```

The SQL either created by the NDT LookML or directly in the "sql:" parameter

customer_order_facts

customer_id	first_order_date	lifetime_amount
1	2015-01-01	300.00
2	2015-01-03	150.00
...	...	...

The resulting derived table

Image reference & Reading
https://docs.looker.com/data-modeling/learning-lookml/derived-tables

Example of NoSQL

We deal with collections of JSON documents
{ <key>:<value>} pairs


```
{
  "_id": "5ffbff2a74fa860c000edb3f",
  "time": "2015-01-01T04:00:00.000Z",
  "amount": 100,
  "customer_id": 1,
  "order_id": 1,
  "_created": "2021-01-11T07:32:58.635Z",
  "_changed": "2021-01-11T07:32:58.635Z",
  "_createdby": "interactivedevict@gmail.com",
  "_changedby": "interactivedevict@gmail.com",
  "_version": 0
}
```

```
{
  "_id": "5ffbff3d74fa860c000edb46",
  "time": "2015-01-01T18:00:00.000Z",
  "amount": 200,
  "customer_id": 1,
  "order_id": 2,
  "_created": "2021-01-11T07:33:17.019Z",
  "_changed": "2021-01-11T07:34:25.037Z",
  "_createdby": "interactivedevict@gmail.com",
  "_changedby": "interactivedevict@gmail.com",
  "_version": 1,
  "_recent_changed": true
}
```


Adapted from RESTdb

order_id	time	amount	customer_id
1	2015-01-01 12::00	100.00	1
2	2015-01-02 02::00	200.00	1
3	2015-01-03 08::00	150.00	2

Let’s Discuss #Databases

#CA

Database Discussion Challenge (GROUP)

In this activity, your team(<=4) will pick a topic and create a slide deck to share with the class.

- Wait for tutor to form breakout rooms.
- Include all team members names in deck + comments **5 min VIVA after 30 mins :)**
- Slide deck of ~5 slides

Topics

1. Uses of NoSQL DB
2. How to use Databases for Virtual Reality
3. Advantages of **SQL DB**
4. Advantages of **NoSQL DB**
5. Disadvantages of **SQL DB**
6. Disadvantages of **NoSQL DB**

Submit as Powerpoint titled “DDAWk01-teamname-yourtopic.pptx”
E.g DDAWk01-Squids-AdvantagesofNoSQL.pptx
*“Leader” - Submit the file in **Google Classroom***

30 minutes

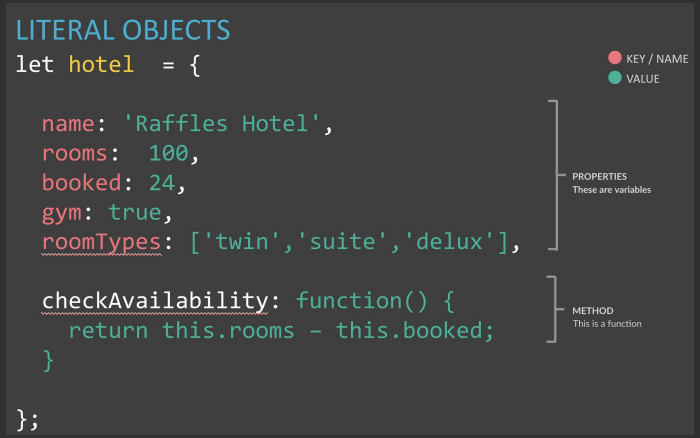
REMEMBER?

#JSON

https://replit.com/@maltester/SimpleJson

JSON looks like object literal syntax but it is just **data**, not an object:

```
{
  "name": "Pikachu",
  "hp": 200,
  "attack": 999,
  "defense": 999,
  "type": "electric"
}
```



https://www.w3schools.com/js/js_json_intro.asp
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON

JSON data is made up of keys and values:

```
{
  "name": "Pikachu",
  "hp": "200",
  "attack": "999",
  "defense": "999",
  "type": "electric"
}
```

KEY in double quotes
Unlike your typical literal objects

VALUE

In JSON, the **key** should be placed in **double quotes** (not single quotes).

The key (or name) is separated from its value by a **colon**.

Each key/ value pair is separated by a comma. However, note that there is *no* comma after the last key/value pair

JavaScript Object Notation

- **JSON is purely a string** with a specified data format — it contains only properties, no methods.
- JSON requires **double quotes** to be used around strings and property names. Single quotes are not valid other than surrounding the entire JSON string.
- **Even a single misplaced comma or colon can cause a JSON file to go wrong,** and not work. You should be careful to validate any data you are attempting to use (although computer-generated JSON is less likely to include errors, as long as the generator program is working correctly). You can validate JSON using an application like JSONLint.
- JSON can actually **take the form of any data type** that is valid for inclusion inside JSON, not just arrays or objects. So for example, a single string or number would be valid JSON.
- Unlike in JavaScript code in which object properties may be unquoted, in JSON only quoted strings may be used as properties.

Reference
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON
https://www.json.org/json-en.html

Simple Player Class that stores the data position and health

JSON Utility
https://docs.unity3d.com/ScriptReference/JsonUtility.html

//JSON Output

```
{
  "position":{"x":5.0,"y":0.0,"z":1.0},
  "health":80
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SimpleJson : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

        Debug.Log("SimpleJson started");

        Player player = new Player();
        player.position = new Vector3(5, 0, 1);
        player.health = 80;

        //converting data from an object to a string
        string json = JsonUtility.ToJson(player);
        Debug.Log(json);

        //parsing back json string to object
        Player loadedPlayer = JsonUtility.FromJson<Player>(json);
        Debug.LogFormat("\nPlayer Data: Position {0} // Health {1} ",
            loadedPlayer.position, loadedPlayer.health);


    }

    private class Player
    {
        public Vector3 position;
        public int health;
    }
}
```

Let's Start

#JSON

#DEMO in C#

#CA

JSON Challenge (GROUP)

In this activity, you will craft how a **JSON** structure can look like in a **game**

Choose a part of the game
Level System, Inventory, Players, Enemy, Stats, Game Mechanics, Gamification, Dialogue, Chat/Messaging, Leaderboard, etc

1. Based on your proposed features, craft your structure (think deeper) and simulate some data
2. You may use the following resource to help you

https://jsoneditoronline.org/

CA GOLD: Create a C# script to handle your json

Submit a JSON titled “DDAWk01-studentid-studentname.json”
E.g DDAWk01-16012010D-SeongGiHun.json
“Leader” - Submit the file in ***Google Classroom***
List team members in the comment

20 - 30 minutes

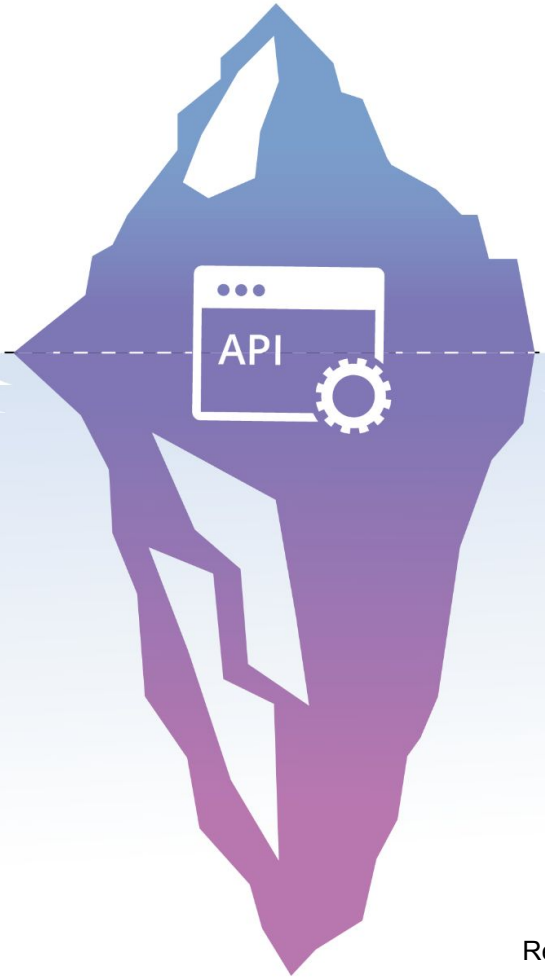
So, what the ***

#FIREBASE

FIREBASE aka BaaS (Backend as a Service)

Frontend
(Developer builds)

- User interface
- Client-side logic



Backend
(Vendor provides as a service)

- Database management
- Cloud storage
- User authentication
- Push notifications
- Hosting

Reading Refence https://blog.back4app.com/backend-as-a-service-firebase/

Why Firebase

The Many Whys

Cloud Based + Managed service + Security backed by Google
Dynamic range of services in Firebase - Auth, Hosting, Databases, etc
Analytics - Insights on how your app is performing
Crashlytics - Figure out why your app is crashing and where
Authentication - OAuth, Google, Facebook, Social Media
Scalable - Able to scale to size as you grow due to Google Infrastructure
Storage Hosting, secure file storage
Sync Real-Time Data - Allows sync across multiple devices)

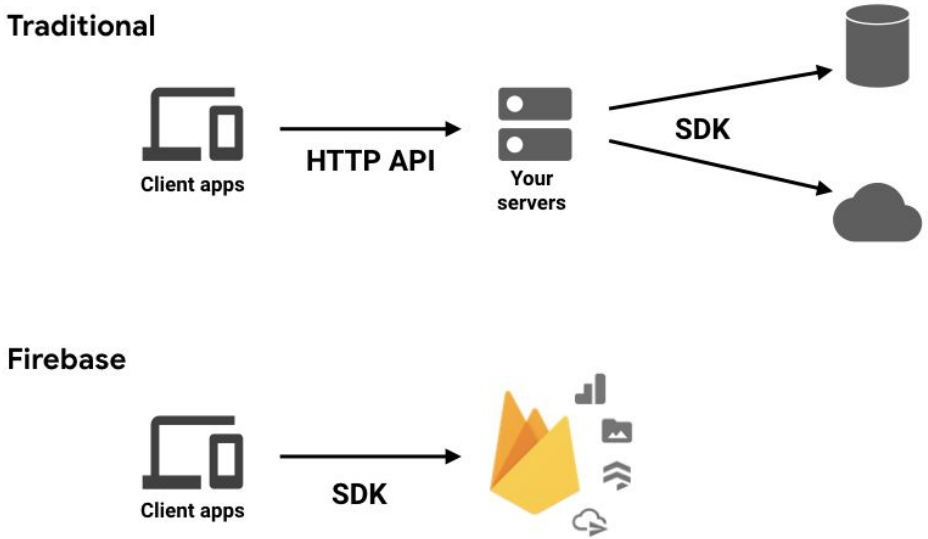
Traditional vs Firebase

Hosted in the cloud - Maintained and operated by Google

Client SDKs are used to interact with Firebase services directly

Traditional Development requires you to work with both frontend and backend software

With Firebase, backend is bypassed and the focus is on the Client interacting with Firebase services based on the administrative rights given



Firebase Pricing

We will be using the Spark plan

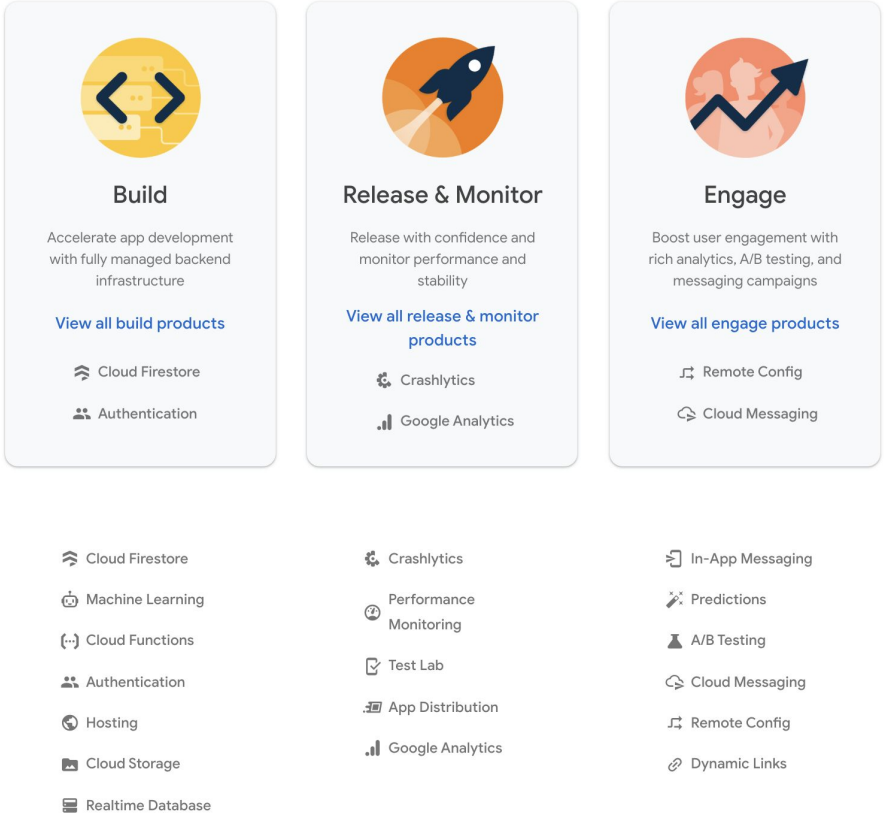
Products	Free Spark Plan Generous limits to get started	Pay as you go Blaze Plan Calculate pricing for apps at scale ✓ Free usage from Spark plan included*
Realtime Database		
Simultaneous connections	100	200k/database
GB stored	1 GB	$5/GB
GB downloaded	10 GB/month	$1/GB
Multiple databases per project	✗	✓
Remote Config	Free	
Cloud Storage		
GB stored	5 GB	$0.026/GB
GB downloaded	1 GB/day	$0.12/GB
Upload operations	20K/day	$0.05/10k
Download operations	50K/day	$0.004/10k
Multiple buckets per project	✗	✓

https://firebase.google.com/

Products	Free Spark Plan Generous limits to get started	Pay as you go Blaze Plan Calculate pricing for apps at scale ✓ Free usage from Spark plan included*
A/B Testing	Free	
Analytics	Free	
App Distribution	Free	
App Indexing	Free	
Authentication		
Phone Auth - US, Canada, and India	10k/month	$0.01/verification
Phone Auth - All other countries	10k/month	$0.06/verification
Other Authentication services	✓	✓
Cloud Firestore		
Stored data	1 GiB total	Free up to 1 GiB total Then $0.108 per additional GiB
Network egress	10 GiB/month	Free up to 10 GiB/month Then Google Cloud pricing
Document writes	20K writes/day	Free up to 20K writes/day Then Google Cloud pricing
Document reads	50K reads/day	Free up to 50K reads/day Then Google Cloud pricing
Document deletes	20K deletes/day	Free up to 20K deletes/day Then Google Cloud pricing

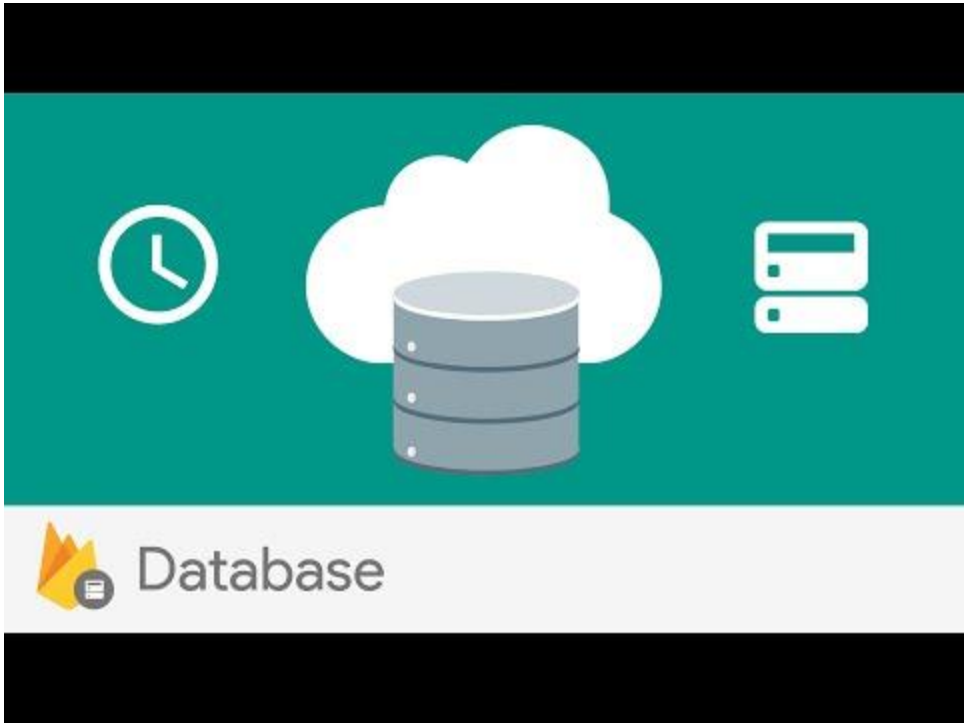
What we are in for

Authentication — user login and identity “Federated Identity”
Realtime Database — Realtime, cloud hosted, NoSQL database
Cloud Storage — massively scalable file storage
Analytics — Numbers behind the game



List of Firebase services

Firebase Real Time Databases



Introducing Firebase Realtime DB
https://youtu.be/U5aeM5dvUpA

SDK vs API

SDK Software Development Kit

It's a set of software tools and programs used by developers to create applications for specific platforms.

- Compiler: Translates from one programming language to the one in which you will work
- Code samples: Give a concrete example of an application or web page
- Code libraries (framework): Provide a shortcut with code sequences that programmers will use repeatedly
- Testing and analytics tools: Provide insight into how the application or product performs in testing and production environments
- Documentation: Gives developers instructions they can refer to as they go
- Debuggers: Help teams spot errors in their code so they can push out code that works as expected
- Java SDK, iOS SDK, Firebase SDK, Facebook SDK

API Application Programming Interface

APIs enhance both the development experience and the end-user experience by adding specific features

- Connecting disparate software applications for a stronger overall product offering.
- Shortening the development cycle through automation.
- Reducing resources that would otherwise need to be allocated for in-house work.
- Improving brand recognition and trust.
- Providing new services to end-users with maximum efficiency.

Rules of SDK vs API

- SDKs usually contain APIs, but APIs don't contain SDKs
- SDKs enable developers to create apps and work as the building blocks for your software
- APIs enable certain functions of applications within the parameters of the SDK with which they're bundled.

	API	SDK
Purpose	Connects and integrates software	Contains a variety of development tools
Characteristics	Lightweight, fast, usually specialized	More robust, usually includes many utilities
Use Case	Used for adding specific functions to an application	Used for creating new applications or adding many functionalities with one package

SDK Explained Simply

SDKs Explained https://www.youtube.com/watch?v=676FMfkYxOk

API Explained Simply

APIs Explained https://www.youtube.com/watch?v=6STSHbdXQWI

Let's Start

#FIREBASE

#DEMO

#CA

Firebase - Write Data (SOLO)

In this activity, we tap on how to write data to our Firebase Realtime Database

1. Follow through the setup guide
2. Based on your suggested JSON structure, adopt a simplified structure and create the appropriate C# scripts and Unity UI to **write** to Firebase Firebase Realtime Databases

CA GOLD: Create additional data to be stored (e.g Leaderboard)

Deliverables

.JSON export from Firebase
Export Package of your scripts
Video clip of your Firebase code in action (show firebase before and after when you run)

Before Next Class

Reference

https://firebase.google.com/docs/database/unity/save-data

Setting Up Firebase

Refer to Guide for extended version
Install Unity Modules

Official Firebase Unity Guide
https://firebase.google.com/docs/unity/setup

Summary

- NoSQL vs SQL database
- What is JSON
- How JSON works in C# & Unity
- API vs SDK
- What is Firebase
- Getting Started with Firebase Real Time Databases and Unity

WEEK 2

NEXT WK HOME BASED!

Lesson Mode SYNCHRONOUS MS TEAMS

READING

https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0
https://www.ibm.com/cloud/blog/sdk-vs-api
https://rapidapi.com/blog/api-vs-sdk/
API vs SDK What's the Diff: https://www.youtube.com/watch?v=kG-fLp9BTRo&t=3s
Working with JSON https://www.nylas.com/blog/the-complete-guide-to-working-with-json/
How do NoSQL databases work https://www.youtube.com/watch?v=0buKQHokLK8
SQL vs NoSQL or MySQL vs MongoDB https://www.youtube.com/watch?v=ZS_kXvOeQ5Y