Excel

1. History of Excel
- Excel was first released in 1985.
- 1985 Excel was launched for the Apple Macin tosh.
- 1987 Microsoft released the first windows version of excel (excel 20 for windows).
- The Latest versions are part of microsoft 365 (formely office 365).
- Excel store file extension in .xlsx, excel support 10,48,576 rows and 16,384 cloumns.

2. Why Need Of Excel ?
- The need of an Excel Sheet arises because it helps people organize, analyze and visualize data easily and efficiently.

Main Reason why Excel is Needed.

a. Data Organization:-
- Excel stores large amount of data in rows and columns .
- Easy to sort , filter, and categorize information(like names, sales , maeks etc.).

b. Automatic Calculations:-
- Perform mathematical and logical operations using formulas. ex- add, subtract, find, average, percentage etc.

c. Data Analysis:-
- Excel help analyze data using pivot table charts and conditional formating.
- Lets users find trends, patterns, summarizes.

d. Data Visualization:-
- You can create charts, graphs and dashboards to make easier to understand.

3. How Excel Works
- Rows -> 1, 2, 3, ...
- Columns -> A, B, C, ...

4. Memory Management Of Excel
- Save number in cells -- under the hood .
- It uses structured data models in, memory compression and binary file formates to manage huge database effectively.
- The entire workbook is loaded in RAM (main memory).
- Excel creates an in Memory Object Model(MOM) that represent worksheets cell, formulas and charts.
- Each elements (cell,row,column), is stored as an Object in memory.
- Excel uses pointers and references to manage relationships between these objects effectively.

5. Data Storage Inside Memory
- Each cell in excel has a small structure in memory containing.
- All the attributes are stored in contigious memory blocks for faster access.
- Excel internally uses a space array model meaning only non-empty cells occupy space in memory.
- Empty cells don't consume significant memory.

6. Type Of Memory Used
- Excel primarily uses RAM(Random Access Memory) when working with files.

CHAKRESH

i. **RAM (main memory)**
   ● Stores entire work sheet structure during use. First read , write.

ii. **Virtual Memory(Paging File)**
   ● Used when data exceeds available RAM(slower).

iii. **Cache Memory(CPU)**
   ● Temporary storage for formulas, functions and re-calculations.

iv. **Disk Storage File Systems**
   ● When saving excel writes data to disk as .xlsx files.

Structured, Semi-Structured and Non-Structured Data

1. Structured Data : Data organized in fixed form rows and clumns like a table colimns fixed in schema. Schema(data types, columns is preferred).
Example:- Excel(tables), MySQL(databases)), CSV(when properly formatted) managed by RDBMS.

2. Semi-Structured Data : Data has some structure but is flexible schema may vary or ot be fixed.
Example:- JSON, XML, CSV(irregular), Excel(with mixed cells)
Data has some structure but not strictly tabular - elements can vary in shape, number or nesting.
Handle by NoSQL- stored arrays, lists and object directly.

3. Non-Structured Data : Data without predefined structure or format - can not fit into row/cloumns easily.
Example:- Text files, text documents, images, emails, audio, video, social media post, word docs, notes
File based on Object - based storage
Hadoop -> HDFS
Amazon s3
Google Cloud storage / Azure blob storage
Tools:- Hadoop, Apache Flink, Elastic Search, Power BI

CSV

1970 -> 1979 IBM - Fortran

1. History(When Released) CSV : Before CSV file exchange with specific elements(commas, tabs or pipes) - Problem:- Memory & Storage limited , Database expensive than user move on comma separated text(value).

2. Why need of CSV : Plan text file format , data store in row and column, each line equal one row of data, each value equal separated by comma(,) , simplicity, universally compatiable, platform independent, fast & lightweight, good for automation, easy to process by code.
Purpose machine readable.

3. Memory Management of CSV : It is just a plan text file -> a continous sequence of bytes stored on disk. Each character -> 'n','a','m','e' stored in ASCII value(or UTF-8) in file.
Chunks --> to RAM
   a. **File Buffer(Temporary) :-** Allocate buffer in loop at runtime chunks(4kb to 8kb) , stored in heap.

CHAKRESH

b. **Parse Data Structure :-** Allocate dynamic memory allocated in heap objects.
actual data --> all go to the heap(stored in arrays) .

small variables --> counter file handle -> stack.

Parsing consumes memory :- Don't load the full csv into memory process on line at a time.

4. Why need csv if we have already excel : As a plain text format for exchangeing tabular data between programs, computer and systems. No standard GUI . Universal text based format.

5. Difference between CSV and Excel :

CSV	Excel
open any text editor ,	need excel or parse
simple rows column ,	complex sheets
small size ,	large size
universally compatiable ,	limited excel software
stored in text,	support rich data
easy to parse(C/C++/Py/Java) ,	harder to parse
error recovery easy	hard to error recovery

JSON

JavScript Object Notation
1. Introduction : This structure is simple, human-readable and easy for computers to parse. JSON introduced in 2000 - 2001 . Douglas Crockford --> Yahoo (SDE) . JSON is lightweight data interchange forms. Application/JSON. AJAX(Asynchronous JavaScript and XML).

2. Why need of JSON :
Need of Data exchange between systems.
Universal standard for APIs & web communiction.
Restful API(Server <--> client communication).
Before json, web services and browser needed ti a way to exchange data-especially for dynamic web applications.
Data using XML, Hunder,drowbacks json provieds a simple, light weight structure for represting sttuctured data format to which human and machines can easily understand.

3. Simplicity and Light Weight:
parse json string --> objects
convert objects --> json strings
json become a universal standard APIs and web communication.
Restful APIs
Configuration
Databases

5. Memory Management of JSON:
Sequence of unicode that represnts structured data.
Just a string in memory.

CHAKRESH

String Array in --> Memory Object
Stored Array bytes in memory
JSON parsing
Garbage Collection
Streaming and DOM parsing
JSON serialization
Performance Consideration
JSON in Cache Memory
Parse and runtime do

4. Why need JSON if we have already CSV.
Json handle nested and Hirericahl data
csv is strictly(2D table only)
Machine readable and self descriptur
Used in web services and APIs
most web services (like Rest APIs) exchange data in JSON , not CSV.
Langauge Independance
Data Type Support

CHAKRESH