

# CSCSE 315

---

TEACHING ASSISTANT: JAY CHEN

JAYCHEN@CSE.TAMU.EDU

# Project 1: Database Management System

---

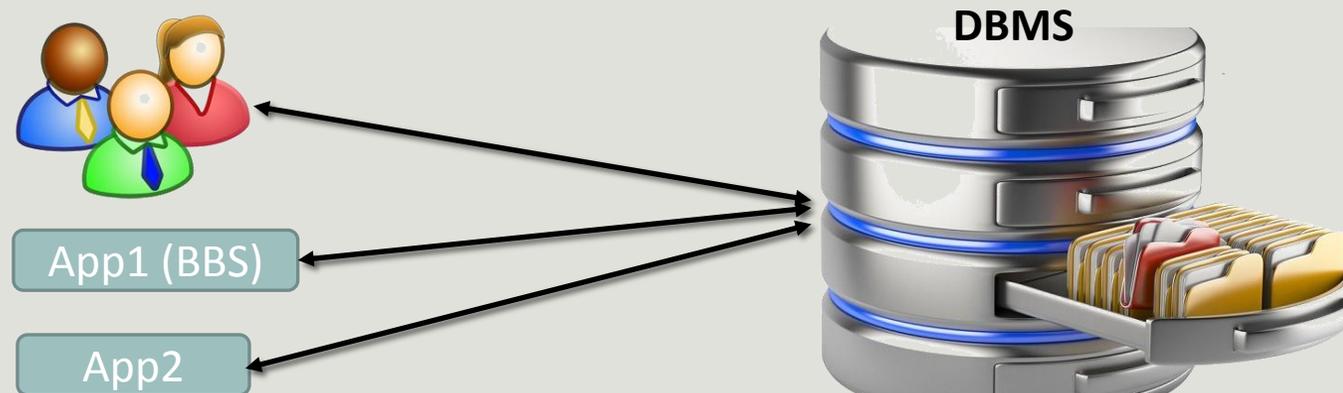
- Design Document
- DB Engine
- Lexical Parser
- DB Engine and Lexical Parser Integration
- DB Application and integration

# Project 1 – Part 2: Database Engine

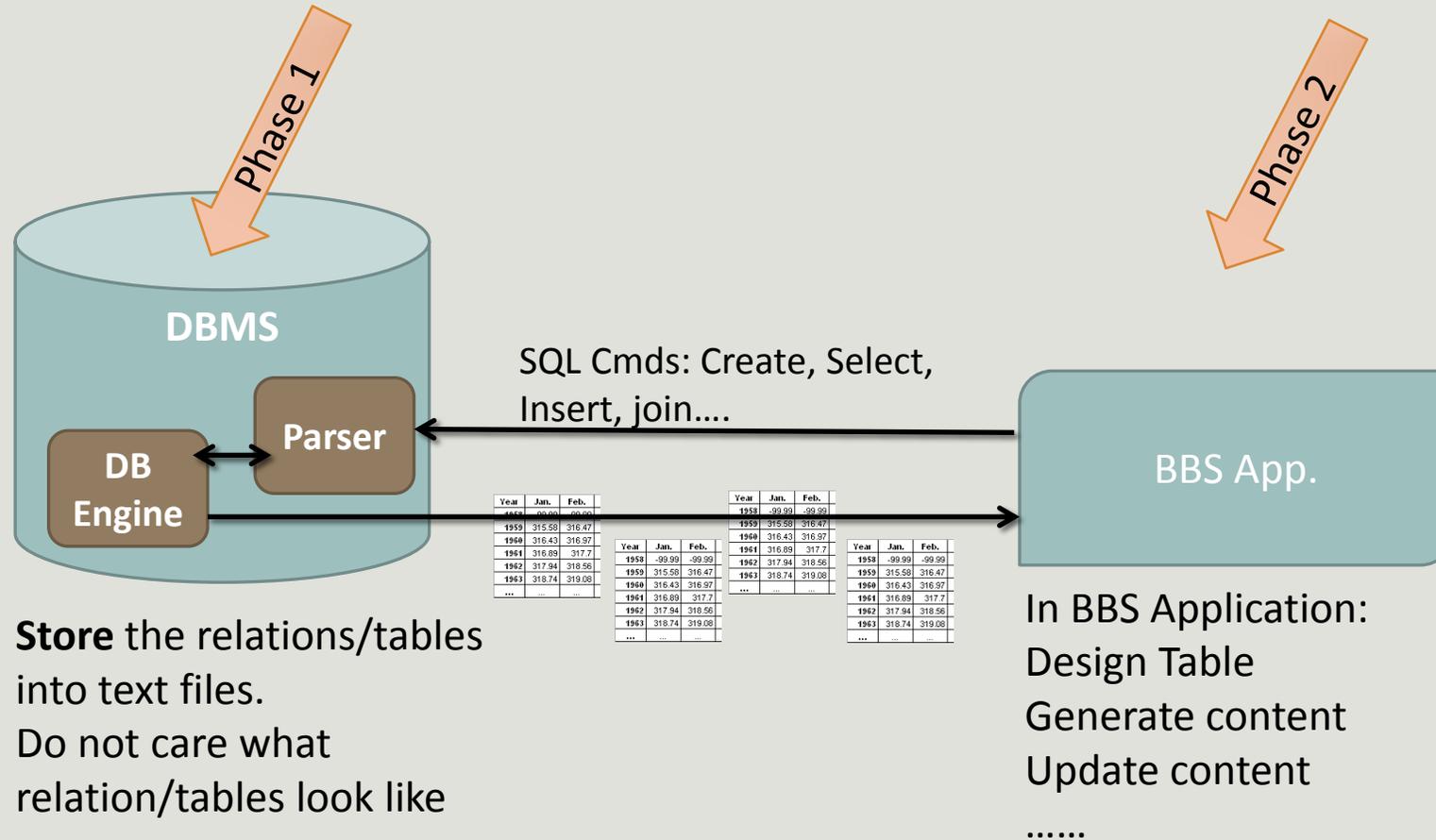
---

- **Objectives:**

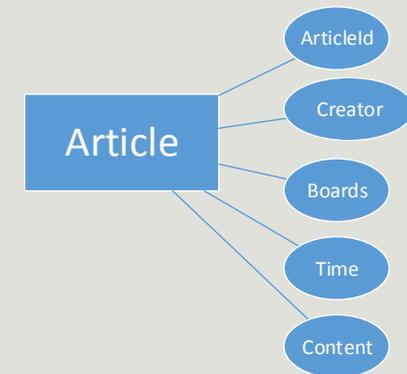
- An engine that can store user data represented by table/relation
- An engine that can return data in table/relation format according to the user's queries
- An engine that allows users to edit (e.g., insert, delete...) the stored table/relation



# Project 1



Year	Jan.	Feb.	March	...
1958	-99.99	-99.99	315.71	...
1959	315.58	316.47	316.65	...
1960	316.43	316.97	317.58	...
1961	316.89	317.7	318.54	...
1962	317.94	318.56	319.69	...
1963	318.74	319.08	319.86	...
...	...	...	...	...



# DB Engine

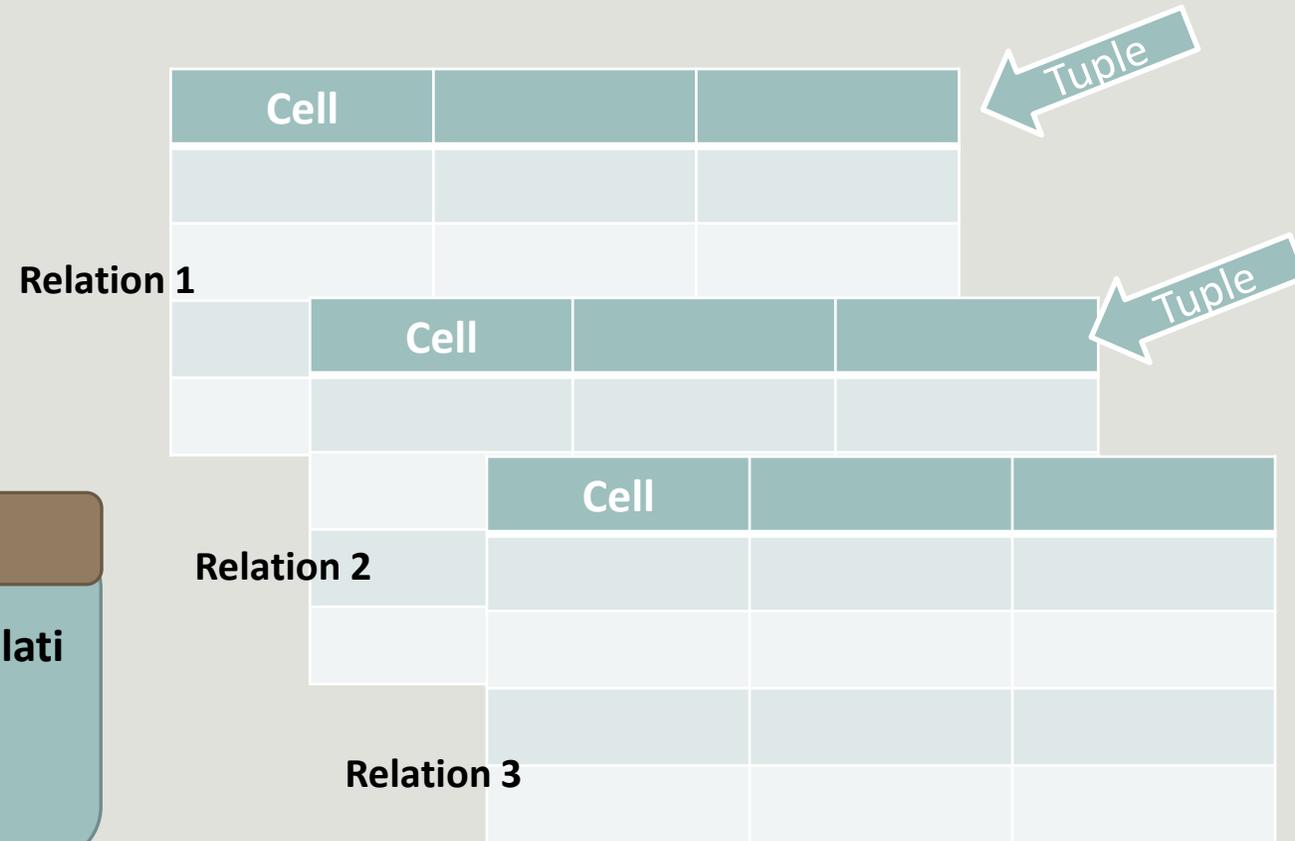
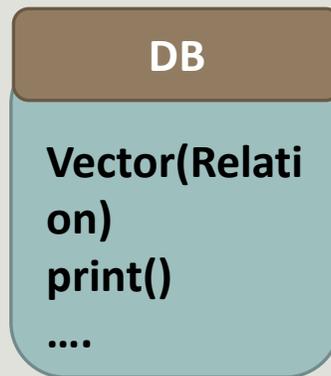
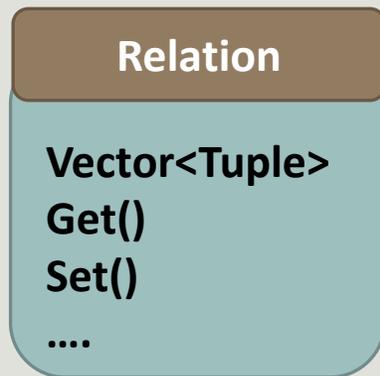
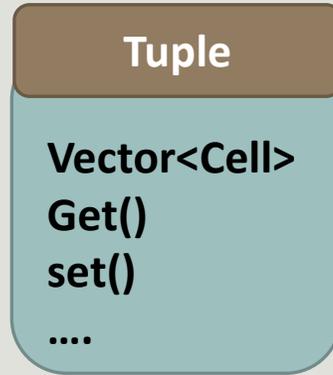
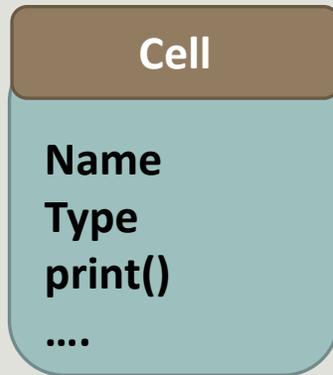
---

## **Some engineering decisions:**

- How to store table in memory? What data structure to use?
- How to store table into text file?
- How to edit/manipulate tables in memory?
- What are the operations the DB Engine needs to support?
  - Selection, Projection, Renaming, Set Union, Set Difference, Cross Product
- What are the commands the DB Engine needs to support?
  - open, close, save, exit, show, create, update, insert, delete

# How to store table in memory?

A simplified example:



# How to store table into text file

---

```
1 CREATE TABLE animals (name VARCHAR(20), kind VARCHAR(8), years INTEGER) PRIMARY KEY (name, kind);
2 INSERT INTO animals VALUES FROM ("Joe", "cat", 4);
3 INSERT INTO animals VALUES FROM ("Spot", "dog", 10);
4 INSERT INTO animals VALUES FROM ("Snoopy", "dog", 3);
5 DELETE FROM animals WHERE name=="Joe"
6
```

- Data must be stored in ASCII text file as shown above (animal.db)
- When you **OPEN()** a relation, the parser reads this file line by line, interprets the commands, and creates the table in memory
- When a client (BBS Application) issues a command that modify this table (e.g., create, insert, delete...), this command has to be stored into this .db file
- This .db file essentially remembers all commands that an application issued before. DB Engine uses these commands to recreate the table when Open()

# Operations that DB Engine supports?

---

- **Create cmd:**

- create a new table in memory

- **Open cmd :**

- open a text file corresponding to the table/relation, parse the SQL script, create a table in memory, and load data into the table
- This function is not required in DB Engine because it needs a working parser.

- **Save cmd :**

- save all the commands to the db text file (don't need to save queries)

- **Show cmd :**

- display/print the table currently stored in the memory

# Operations that DB Engine supports?

---

- **Insert query:**

- add a tuple to a table/relation in the memory

- **Delete query:**

- remove a tuple from a table/relation in the memory

- **Select query:**

- search and return one more tuples from a table in the memory

- **Project query:**

- Return a subset of attributes (columns of the table)

- **Product query (Cross-Join) :**

- Each row in the first table is paired with all the rows in the second table

- **Union-Compatible:**

- Two relations are union-compatible if they have the same # of attributes and each attribute must be from the same domain