1. What is a System Schema?

A **System Schema** is a built-in schema in SQL Server (or other RDBMS) that contains **system-defined objects** used to manage and operate the database engine.

❖ **Key Points:**

- System schemas include objects like system views, functions, procedures, and tables.
- Common system schemas in SQL Server:
  - sys: contains catalog views to access metadata (e.g., sys.tables, sys.objects)
  - INFORMATION_SCHEMA: provides standardized views for metadata (e.g., INFORMATION_SCHEMA.TABLES)
  - db_owner, db_accessadmin, etc. — special roles associated with permissions.

❖ **Use Cases:**

- Query system metadata (sys.tables, sys.columns, sys.indexes, etc.)
- Monitor performance, structure, and relationships between database objects

➢ **Example:**

```
SELECT name FROM sys.tables; -- returns all table names in the current database
```

---

2. What is Procedure Cache and Cold Cache?

❖ **Procedure Cache (or Plan Cache):**

- The **Procedure Cache** stores **execution plans** of previously run queries or stored procedures.
- It saves time by **reusing** compiled plans instead of re-compiling on each execution.

❖ **Benefits:**

- Faster execution (no need to re-compile query plans)
- Reduces CPU usage
- Improves performance for repeated queries

➢ **Example:**

```
SELECT * FROM sys.dm_exec_cached_plans; -- shows cached execution plans
```

❖ **Cold Cache:**

- A **Cold Cache** refers to a situation where the **procedure cache and/or data cache is empty or flushed**.
- This happens after:
    - Server restart
    - Cache is cleared manually
    - First run after deployment

❖ **Result:**

- Queries will take longer to execute initially because:
    - Execution plans must be recompiled
    - Data pages need to be read from disk (not memory)

❖ **Used for:**

- Performance testing in realistic scenarios
- Benchmarking queries without memory bias

➢ **Example:**

```
DBCC FREEPROCCACHE; -- clears the procedure cache

DBCC DROPCLEANBUFFERS; -- clears data cache (cold data)
```