SQL Server Data File Architecture

Component	Description	Details
Data Files (.mdf, .ndf)	Physical files that store the data and objects of a SQL Server database.	- **Primary Data File (.mdf):** The main data file containing startup information for the database. - **Secondary Data Files (.ndf):** Optional files to spread data across multiple disks.
Filegroups	Logical grouping of data files within a database, allowing easier management and optimization.	- **Primary Filegroup:** Contains the primary data file (.mdf) and objects not assigned to other filegroups. - **Secondary Filegroups:** Used for partitioning or to distribute large tables and indexes across multiple files.
Extents	A collection of eight 8 KB pages, totaling 64 KB, which SQL Server uses to manage space allocation in data files.	- **Uniform Extents:** All eight pages belong to a single object. - **Mixed Extents:** Pages can belong to different objects, used for small tables or indexes.
Pages	The fundamental unit of data storage in SQL Server, each 8 KB in size.	- **Types:** Data Pages (storing table data), Index Pages, GAM/SGAM Pages (managing extents), PFS Pages (tracking page usage). - **Page Header:** Contains metadata like page ID, type, and status.
GAM and SGAM Pages	Specialized pages that manage and track the allocation of extents in the data files.	- **GAM (Global Allocation Map):** Tracks which extents are allocated. - **SGAM (Shared Global Allocation Map):** Tracks mixed extents that have free pages.
PFS Pages (Page Free Space)	Pages that record the allocation status and free space availability of individual pages within an extent.	- **Purpose:** Used to quickly find free space within an extent. - **Frequency:** A PFS page occurs every 8,088 pages (approximately 64 MB).
IAM Pages (Index Allocation Map)	Pages that map the extents used by a table or index, helping SQL Server manage space allocation for objects.	- **Purpose:** Tracks which extents belong to a table or index. - **Multiple IAM Pages:** Required for large tables or indexes spanning multiple filegroups or extents.

LOB Data Pages	Pages used to store large object (LOB) data types such as `text`, `ntext`, `image`, `varchar(max)`, `nvarchar(max)`.	- **Storage:** Stored separately from regular data pages if the data exceeds 8 KB. - **Pointers:** Regular pages store pointers to LOB data pages.
Row-Overflow Data	Handles variable-length columns that exceed the 8 KB page size limit.	- **Overflow Pages:** Data is moved to these pages when it exceeds the 8 KB limit. - **Pointer:** The original page stores a pointer to the row-overflow page.
Partitions	Logical divisions of a table or index across multiple filegroups, improving manageability and performance.	- **Purpose:** Helps manage large tables by distributing them across filegroups. - **Partitioning Key:** Determines how the data is divided.
Sparse Files	Files that only allocate disk space as data is written, reducing the initial size of the file.	- **Purpose:** Saves space when the actual data stored is sparse. - **Use Case:** Typically used for columns that have a lot of NULL values.
Instant File Initialization	A feature that allows SQL Server to skip zeroing out space when creating or growing data files, speeding up operations.	- **Benefits:** Speeds up file creation and growth by not initializing the file with zeros. - **Requirement:** Requires specific Windows permissions (Perform volume maintenance tasks).
File Growth	The automatic expansion of a data file when it reaches its allocated size.	- **Settings:** Can be configured to grow by a fixed size or by a percentage. - **Impact:** Uncontrolled growth can lead to fragmentation or excessive VLFs in the transaction log.
Shrink Operations	The process of reducing the size of a data file by removing unused space.	- **Consideration:** Frequent shrinking can cause fragmentation and impact performance. - **Best Practice:** Should be used cautiously and typically avoided unless necessary.
File Stream	A feature that allows large binary data (e.g., documents, images) to be stored directly on the file system instead of in the database.	- **Integration:** Managed through T-SQL but stored outside the database files. - **Use Case:** Ideal for storing large unstructured data with transactional consistency.

Max Size	The maximum size a data file can reach, configurable during file creation or through later modifications.	- **Settings:** Can be set to a fixed maximum or to unlimited (auto-grow until disk is full). - **Consideration:** Proper planning is required to avoid running out of space unexpectedly.
TempDB Architecture	The system database used for temporary objects, intermediate query results, and version stores.	- **Files:** Usually has multiple data files for better performance in multi-core systems. - **Rebuilt on Startup:** TempDB is recreated every time SQL Server restarts, clearing all temporary data.
Buffer Pool	The area of memory where SQL Server caches pages read from the database files.	- **Purpose:** Reduces the need for disk I/O by keeping frequently accessed data in memory. - **Lazy Writer:** Process that periodically writes dirty pages back to disk.

This table provides a comprehensive overview of SQL Server Data File Architecture, covering the structure, management, and operational aspects of data storage within SQL Server.