Pre-requisites

09 September 2022 08:34

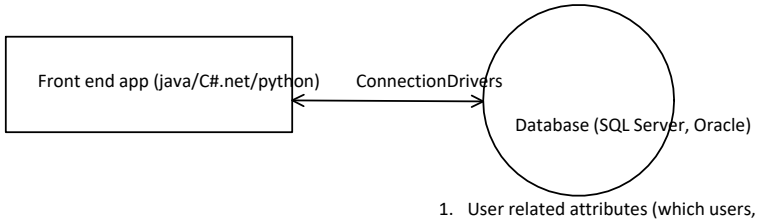
Lab Setup Requirement	Hardware CPU - Intel Core i3/i5/i7 processor, RAM - at least 8 GB HDD-512 GB / 1 TB, OS - Windows 10 /8.1/11, MS Word/excel, PowerBI desktop (optional)

Software - SQL Server 2016, 2017 or 2019 Enterprise/Developer edition, Visual Studio 2019/2022/VS code, Azure CLI, Storage explorer, SQL Server Management Studio/Azure Data Studio, Microsoft Azure Subscription, Git tools and GitHub account, Azure PowerShell, PowerShell ISE, AWS Tools for VS code, GCP Tools for VS code.

Database Fundamental & SQL Server BI 2016

08 September 2022 18:45

Application metadata



1. User related attributes (which users, port, MSSQL - 1433, encryption, protocol (TCP)
2. SQLConnection
3. ADOConnection (ADO.Net)

Data Dictionary

1. Names of all of the database tables and their schemas (Sales, Customers, Orders, Employees…)
2. Details of all the tables in the database like owners of the tables, the security constraints, when the tables were created etc.
3. Physical information of the tables in the database - where the tables in the db itself have been stored and how
4. Table constraints includes primary key information, foreign key information etc.
5. Information related to database views which are visible

Employee Table - Active Data Dictionary -- self updating

Passive Data Dictionary -- manually updated to match the database

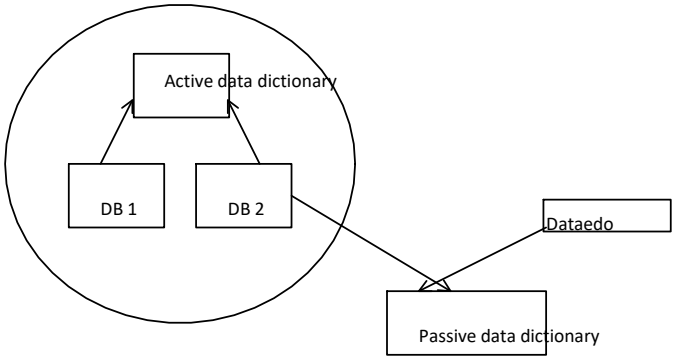
Field Name	Data Type	Field size for display	Description	Example	Dept_id
Employee No	INT	10	Unique ID for the employee	444007	1
Employee Name	VARCHAR(50)	20	Name of the Employee	James Hobb	23

Example of Passive Data Dictionary -

Dataedo -- tools

Column	Data Type	Description
Field Name	10	
Data Type	20	

Dataedo



Different Types of Database

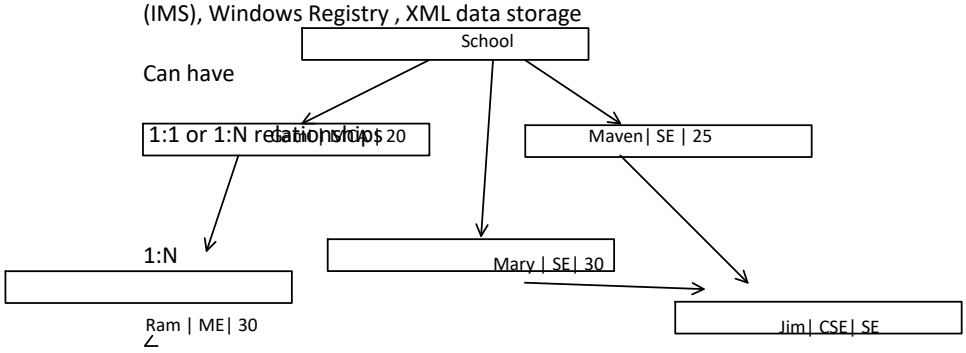
1. Relational Database - consists of set of tables with columns and rows
2. Object-oriented database - information can be presented in the form of objects as in object-oriented programming. Inclined towards into objects e.g. multimedia record in a relational database can be defined as definable data object, MongoDB has offering of Object oriented database
3. Distributed database - consists of two or more files located in different sites, e.g. SQL server mirror databases, distributed dbs

4. Data Warehouses - central repository for data storage, includes a type of database designed for faster query and analysis(SSAS, SSIS)
5. NoSQL databases - non-relational db - support for unstructured, semi-structured data, dynamic schema, flexible and faster data retrieval (Cassandra, Mongo DB, Couch DB, Azure Cosmos DB, AWS Document DB, AWS Dynamo db)
6. Graph Databases - nodes - entity , attribute - relationship

e.g. Apache Tinkerpop , Azure Cosmos db Graph API , Neo4j

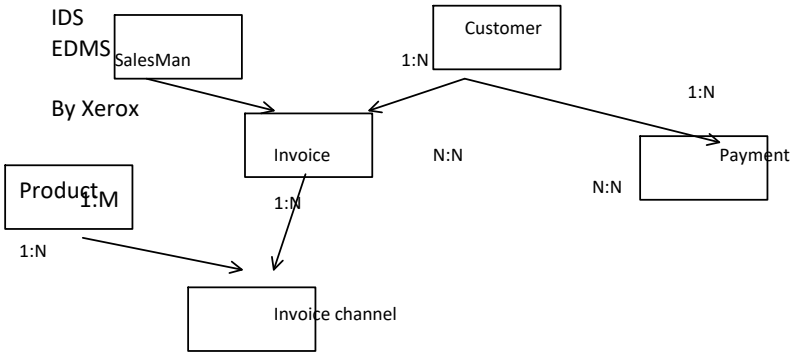
7. Cloud databases - Databases as a service (DBaaS) e.g. Azure SQL database, Azure SQL managed instance
8. Document / JSON database - designed storing, retriving and managing document oriented information. (Azure Cosmos db SQL API, document db, AWS document db, data being stored in key-value pairs,

Hierarchical data model - COBOL (DB2) - IBM Information Management System (IMS), Windows Registry , XML data storage



Network data model

1. An owner record which is the same as of the parent in the hiearchical model
2. A member record which is same of child in the hiearchical mode



Employee Table 1

Fields	Columns(attribute 1) Emp ID	Columns(attribute 2)	Attribute 3
Row 1 (records/tuples)	1001		
Row 2 (records/tuples)	1002		

Employee ID - foreign

Table 2
EmployeeAddress

Fields	Columns(attribute	Columns(attrib	Attribute

Hiearchy Data models

Mainframes DBMS for IBM

IBM IMS and RDM Mobile - embedded db

Employee table

Emp No	First Name	Last Name	Dept
1001	Alan	Turing	Finance

Device table

Serial no	Type	User emp no
001	Monitor	1001

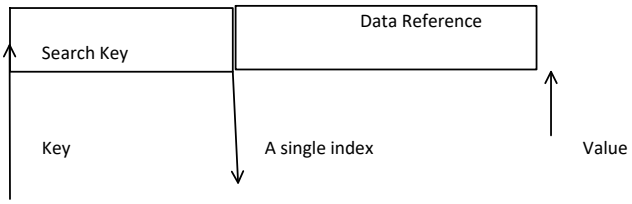
	1) Emp ID	ute 2)	3
Row 1 (records/tuples)	1001		
Row 2 (records/tuples)	1002		

Entity Integrity - ensures the primary key in a table is unique and the value is not set to null

Referential Integrity - requires every value in a specific foreign key column should be found in the primary key of the table from which it is originated.

Index

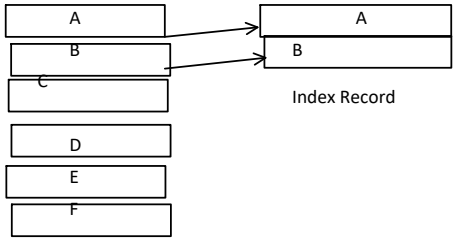
- First column for a table is the Search key which can contain a copy of the primary key of the table. These values are stored in sorted order so that the corresponding data access can be faster.
- The second column is the data reference or Pointer which can contain a set of pointers holding the addresses of the underlying disk blocks where the specific key values are stored.



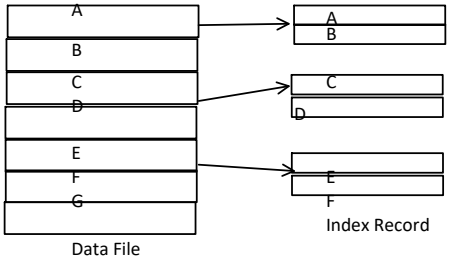
1. Access Types - value based search, range of access over data records
2. Access Time - time required to find the data element
3. Insertion time - time taken to find the specific space and to insert the new data
4. Deletion time - time taken to find an element & to delete it
5. Space overhead - additional space required by an index

File storage mechanism to follow for indexing

1. Sequential file organization -



Dense Index



Sparse Index

Foreign Key

1. The foreign key constraint is used tp prevent actions what would destroy links between tables.
2. A foreign key is a field (collection of fields) on a table refers to primary key in another table.
3. A table with the foreign key is called as child table, and the table with the primary key is called parent/referenced table.

s			

2. Hash file organization - indices are choosen based on values being distributed uniformly. Hash buckets where the value is assigned determined by hash function.

a) Clustered index - you can define an index with upto 16 columns , The max size of this index should be 900 bytes

The columns defining for clustered index is termed as clustering key.

SQL server to order the data in the table in accordance to the clustering key.

a) Non clustered index

- Do not impose a sort order on the table
- Restriction wise max size supported as 900 bytes & can be promoted max 16 columns of a table, max 249 non-clustered indexes can be created on a table.

WHERE clause used to specify the condition while fetching the data from single table or joining from multiple tables. When, a given condition is satisfied, use the WHERE clause to filter the specific records and fetching the necessary records.

Surrogate Key

School A

Reg No	Name	% obtained
201010	Brian	66

Reg No	Name	% obtained
201010	Brian	66
202012	Max	50

School B

Reg No	Name	% Obtained
CS300	Ava	50
DS500	Maria	60

Merging these two tables in a single sql table

1. Automatically generated by the system
2. It hold anonymous integer
3. It contains unique value for all records in the table
4. Value cannot get modified
5. Easier identification purposes

Surr_id	Registration no	Name	% obtained
1	201010	Brian	66
2	202012	Max	50
3	CS300	Ava	50

from single table or joining from multiple tables. When, a given condition is satisfied, use the WHERE clause to filter the specific records and fetching the necessary records.

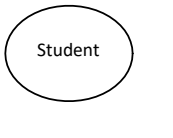
ALTER ID, NAME, SALARY FROM CUSTOMER WHERE NAME = 'XYZ'
DELETE ID, NAME, SALARY from CUSTOMER WHERE NAME = 'ABC'

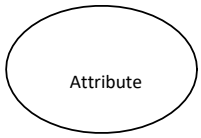
a) Multilevel index

ER Modelling

Entity - A Entity is an object with a physical existence - a particular person, car, house,

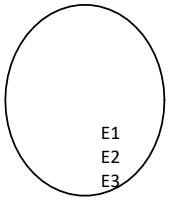
A entity is an object of entity type & set of all entities is called as entity set.

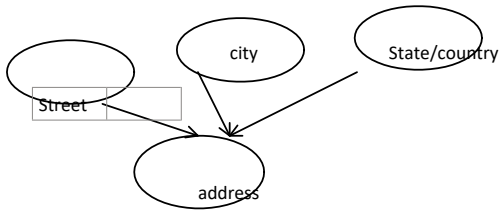


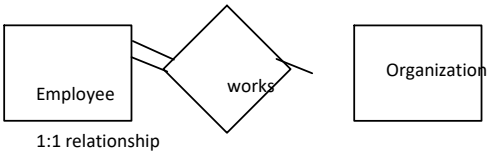


Student_roll_no
Student_name
Student_DOB
Student_Address

Key Attribute -
student_roll_no







1:1 relationship
Students - enrolled to - Courses

M:N M:1

Unary relationship

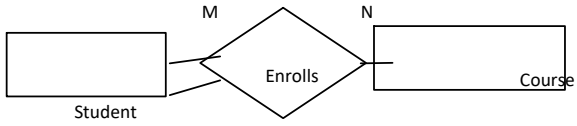
Multivalued attribute

Derived attributes
Student_age,
Employee_bonus

Participation constraint

1. Total Participation - each entity must be participated in the relationship.
2. Partial Participation - entity in an entity relationship may or may not participate in the relationship.

Employees --- Taking leaves -- During vacation (M:N)



PersonId (Primary Key)	LastName	FirstName	Age
1	Bill	Johns	30
2`	Maria	Sophia	21

SQL Command

1. DDL - CREATE, DROP, ALTER, TRUNCATE
2. DML - INSERT, UPDATE, DELETE
3. DCL - GRANT , REVOKE
4. TCL - Commit, Rollback, Savepoint
5. DQL - SELECT

Orders table

OrderID	OrderNumber	PersonID (Foreign Key)
1	77445	2
2	88445	1

The foreign key constraint prevents invalid data from being inserted into the foreign key column, since it has to be one of the contained value in the parent table.

Normalization

09 September 2022 18:33

- Normalization is the process of organizing the data into the database
- Normalization is used to minimize the redundancy from a relation or set of relations. It's also defined to eliminate the undesirable features like insertion, update and deletion anomalies.
- Normalization divides the large unnormalized tables into smaller and links them using relationships.
- The normal form is used to reduce the redundancy from the database level.

Purpose of normalization -

Data modification anomalies can be differentiated into the types:

1. Insertion Anomaly - a new row/tuple cant be inserted into a relationship due to lack of data
2. Deletion Anomaly - The delete anomaly refers to the scenario, where the deletion of data from the row results in loss of some other important data due to lack of proper key based attribute level relationships.
3. Updatation Anomaly - The update anomaly can exist when an update operation of a single data value requires multiple rows/tuples of data to be amended/updated.

	1NF	2NF	3NF	BCNF	4NF	5NF
Conditions	Elimination of repeating of groups	Eliminate the partial functional dependency.	Reduce the transitive dependency	More advanced level than 3NF. More stricter enforced for 3NF.	Eliminates the concepts of multi-values dependency .	Eliminates the concepts of joining dependency
Feature	A relation in first normal form when it consists of only an atomic value. One attribute contains only one value for a specific row.	Tables should be in 1NF + non-key attributes which are fully functional type they must dependent on the primary key	The relation fulfills the criteria for 2NF + no transitive relationship exists.	A table in BCNF, if there is a functional dependency exists x-> y (x is assumed to be the super key). Table should be in 3NF. For every functional dependency, the left side of relationship of the table fulfills the criteria for super key.	The table should in BCNF, it should have no multi-value dependency .	The table should be in 4NF + there cant be any join level dependency exists in the table, joining of the table should not incur any data loss or any joining should not have any particular loss.

Benefits of normalization -

1. Reduce the data redundancy
2. Greater data organization & consistencies.
3. Flexible level of database design
4. Enforce the concepts the referential integrity

First Normal Form (1NF)

- The table should be in the form where the one attribute should contains only one value based on specific row / tuple
- A table should have atomic values (no duplication of values on attributes) , enforce non-repetitive groups/attributes

- The column should have one single valued attribute.

Unnormalized table - Employee table

Employee_id	Employee_name	Employee_Phone	Employee_Email	Employee_HireDate	Employee_Salary
001	Mark	1988233222 111222333	mark@contoso.io Mark.b@fabrikum.com	01/01/2021	4000
002	John	222444111 444333777	john@contoso.com jo@adven.com	03/02/2017	3500

1NF is fulfilled for Employee table

Employee_Id	Employee_Name	Employee_Phone	Employee_Email	Employee_HireDate	Employee_Salary
001	Mark	1988233222	mark@contoso.io	01/01/2021	4000
001	Mark	111222333	Mark.b@fabrikum.com	01/01/2021	4000
002	John	222444111	john@contoso.com	03/02/2017	3500
002	John	444333777	jo@adven.com	03/02/2017	3500

2NF - Second Normal Form

- To be in 2NF, the tables should be in 1st Normal Form.
- All non-key attributes should be fully functional dependent on the primary key of the table

In this table, non-prime attribute Employee_Name is dependent on the Employee_ID which is proper subset of a candidate key.

Employee_detail table

Employee_ID	Employee_Name	Employee_HireDate
001	Mark	01/01/2021
002	John	01/01/2021
003	..	..

Employee_Salary table

Employee_ID	Employee_Salary	
001	4000	
002	3500	

Employee_Contacts table

Employee_ID	Employee_Phone	Employee_Email
001	1988233222	mark@contoso.io
002	111222333	john@contoso.com

Professor table (1NF)

ID	Course	Univ Name	Name
10	CSE	Stanford	
15	IT	Harvard	

15	ME	Harvard	
30	DB	Princeton	
30	CA	Princeton	

2NF

Professor_detail table

ID	Univ Name	Name
10	Stanford	Mark
15	Harvard	Mark
30	Princeton	John

Fulfills the partial dependency.

Professor_subjects table

ID	Courses
10	CSE
15	IT
15	ME
30	DB
30	CA

Third Normal Form (3NF)

- A table can be in 3NF, if it is in 2NF, should not have any partial functional dependency
- It should reduce the data duplication
- It can achive the integrity
- No transitive dependency between non-prime attributes/columns.

  A -> B -> C => Column A is dependent on B , B dependent on C, if A is also dependent on C, then it's called as transitive dependency.

  Employee_details (2NF)

Emp_id	Emp_Name	Emp_Zip	Emp_State	Emp_City
222	Mark	70045	Arizona	Phoenix
333	Harry	34404	Utah	Lake
555	Jerry	40032	Arizona	Maveric
335	Hannah	33406	New Mexico	Titan

  Super Key relation -> (Emp_id), (Emp_id) (Emp_Name), (Emp_id)(Emp_Name)(Emp_Zip)…

  Candidate key -> Emp_id

  Delhi - 11….
  Mumbai - 4….

  Emp_State and Emp_City is dependent on the Emp_Zip & Emp_Zip is dependent on the Emp_id. The non-primary key attribute (Emp_State) and (Emp_City) transitively dependent on primary key (Emp_ID). It violates the criteria of 3NF.

  Employee tbl

Emp_ID	Emp_Name	Emp_Zip

222	Mark	70045
333	Harry	34404
555	Jerry	40032
335	Hannah	33406

Employee_zipcode table

Emp_Zip	Emp_State	Emp_City
70045	Arizona	Phoenix
34404	Utah	Lake
40032	Arizona	Maveric
33406	New Mexico	Titan

BCNF - (Boyce Codd Normal Form)

- It is stricter than 3NF, more advanced than 3NF.
- A table will be in BCNF if every dependency exists like with a super key to no the attribute, x -> y. (x is the super key of the table).

Employee_details table (3NF)

Emp_id	Emp_country	Emp_dept	Depart_Name	Emp_depart_no
333	US	Manufacturing	Design	001
444	UK	Software	Engineering	002
555	US	Architecture	Development	003
666	US	Machinery	Development	004

Functional Dependency ->

Emp_id -> Emp_Country
Emp_Dept -> Department_name, Emp_depart_no

Candidate key -> (emp_id, emp_dept)

This table is not in BCNF, because neither the Emp_Dept and Emp_id alone the keys.

Emp_country table

Emp_id	Emp_country
333	US
444	UK
555	US
666	US

Candidate key - Emp_id

Emp_department table

Emp_dept	Emp_Department_name	Emp_dept_no
Manufacturing	Design	001
Software	Engineering	002
Architecture	Development	003
Machinery	Development	004

Candidate Key - Emp_Dept

Third table should have both of these candidate keys

(Emp_id, Emp_dept)

Emp_dept_mapping table

Emp_Id	Emp_Dept
333	Manufacturing
444	Software
555	Architecture
666	Machinery

The left side of both the functional dependencies is a key. --> This table is in BCNF.

Forth Normal Form (4NF)

- A table is in fourth normal form (4NF), if it's in already in BCNF & has no multivalue dependency.

A -> B, if for single value of A, multiple values of B exists, then it's called as multivalue dependency.

Student table (3NF)

ID	Course_enrolled	Programming_skills
10	Computer Science	C
10	Engineering Math	java
30	IT	Networking
40	CS	Compiler Design
55	Bioinformatics	C

There's multivalue dependency exists for the student with ID=10

Course_enrollment table

ID	Enrolled_courses
10	Computer Science

10	Engineering Math
30	IT
40	CS
55	Bioinformatics

skills

ID	Programming
10	C
10	java
30	Networking
40	Compiler design
55	C

Fifth Normal Form - 5NF

- A table is in 5NF, if it's already in 4NF & should not contain the join dependency. The joining should not incur any data loss.
- 5NF is satisfied when all the table are being broken into as many as tables as possible to avoid redundancy.
- 5NF is called project level join normal form.

Employee table

Employee_Name	Employee_HireDate	Employee_Designation
John	01/01/2020	Sr. Software Engineer
Mark	01/03/2021	Software Arch
Mark	01/03/2021	Software Engineer
Celine	04/02/2014	Sr. Software Engineer
Alan	01/03/2021	Network Engineer

P1 level

Employee_Desig	Emp_Name
Sr. Software Engineer	John
Software Arch	Mark
Software Engineer	Mark
Sr. Software Engineer	Celine
Network Engineer	Alan

P2 level

Emp_Name	Emp_HireDate
John	01/01/2020
Mark	01/03/2021
Mark	01/03/2021
Celine	04/02/2014

Alan	01/03/2021

P3 Level

Emp_Design	Emp_HireDate
Sr. Software Engg	01/01/2020
Software Arch	01/03/2021
Software Engineer	01/03/2021
Sr. Software Engineer	04/02/2014
Network Engineer	01/03/2021

TSQL concepts

08 September 2022 18:45

SQL Table Design

1. Data types : A data type is fundamental constraining element of a database which restricts the range of possible values that are allowed to be stored in a column

a) Numeric data type :
- tinyint (0-255)
- Smallint(-32768 to 32767)
- Int (storage space - 4 bytes)
- Bigint(8 bytes)
- Decimal(p,s) fixed precision and s - scale numbers , precision - max total no of decimal digits can be stored , scale - number of decimal digits which are stored to the right of precision point.
- Numeric(p,s) - a constant data value can be automatically converted to a numeric data value. SQL server uses the default rounding options when converting a number to decimal or numeric one with smaller precision and scale.
- Smallmoney (-214748.00… 214748.00) - 4 bytes
- Money - 8 bytes accuracy of 10000 of monetary units.
- Real -3.4 , -1.18 to positive values (4 bytes)
- Float - 4 bytes or 8 bytes

b) Character data type
- Char(n) - 1 byte / character upto 8k bytes
- Varchar(n) - max 8k bytes
- Text - stores upto 2 gb
- Nchar(n) - 2 bytes per character max 4k bytes
- Nvarchar(n) - 2 bytes per character max of 4k bytes
- Ntext - 2 bytes per character stored upto 2 gb
  The (n) characters defined sets the max no of characters allowed to be stored in the column
  Nvarchar and varchar - the amount of storage consumed is equal to the number of characters being stored.
  Varchar(max), nvarchar(max) - 2 gb of data

c) Binary data type
Binary data type can be fixed length or variable length

- Binary (sizes of column data entries are consistent) upto 8k bytes
- Varbinary - variable length binary data
- Varbinary(max) - storage exceeds beyond 8k bytes
- Image - variable length binary data upto 2 gb
- Alternative varbinary(max)

d) Spatial data type:
- Geography - implemented as .net CLR (latitude, longitude)
- Geometry - store points, lines, curves

e) FileStream data type:
BLOB data stored , not restricted to 2 gb of limit of file system

f) HiearchyID data type:
Storing of nodes & edges/vertices of graphs, flowcharts

SQL Server Column properties

g) Sparse Columns
The attribute of a specific row if requires very small values & need small storage space, then can use the Sparse property.

-- Temporal tables

It is a database feature which brings built-in support for providing the information about the data stored in the table in time, rather than only the data which is correct at the current moment of time.

System-versioned temporal table is kind of user table designed to keep a full history of data changes, allowing for easy point-in time analysis.

Temporal tables have two explicit defined columns with datetime2 data type, these columns are called as period columns.

Benefits

- Auditing all data changes and performing data forensics
- Calculate the data column change trends over the time
- Maintain a slowly changing dimension for the decision support apps
- Recover from accidental data damages and errors

string connectionString = "Data Source:MSSQL1;"+"Initial Catalog=sampleDB;Integrated Security=SSPI;" +
"MultipleActiveResultSets=True"

Session cache -> logical session
SqlClient driver (c#) caches the MARS session within a connection. 10 MARS session.

SQL Server Clauses:

1. SQL Order By Clause:
- Order the result set of a specific query by the specified column list and optionally it also limits the rows returned to a specified range. The order in which the rows are returned in a result set are not been guaranteed unless an ORDER BY clause is defined
- Determine the order om which Ranking function values are applied to the result set. --> Ranking function helps to return a ranking value for each row in a partition.

ORDER BY clause is not supported for CREATE TABLE AS SELECT(CTAS) statements in Azure Synapse & AAS.

ORDER BY expressions
[collate collation_name]
[ASC | DESC]

2. HAVING Clause -
- Specifies the search condition for a group or an aggregate. HAVING clause can be used only with the SELECT statement. HAVING is typically used with the GROUP BY clause. When the GROUP BY clause is not used, there's an implicit single, aggregated group.

[HAVING <search_criteria>] - one or more predicates for groups/aggregates to meet.

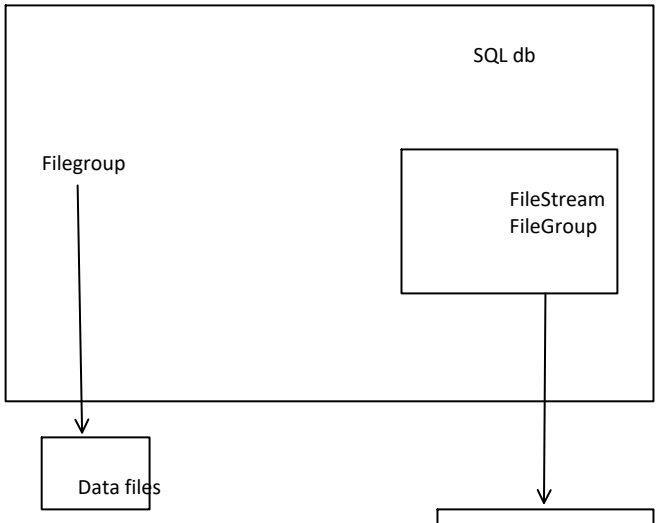
The text, image and ntext data cant work with HAVING Clause.

FileStream in SQL Server

Database structure

- .mdf file - primary database
- .ndf file - secondary db
- .ldf file - transaction log file
-
- Data and log file for SQL server

a) Physical file name Fi
b) Initial file size
c) File growth factor
d) Maximum size
e)
f)



data filestream full-text	Filegroup is responsible to create storage boundary for tables and indexes

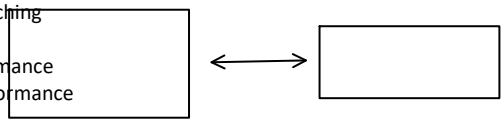
Emp id	Emp_name	Emp_phone (varbinary(max))
1	Alan	0*E98886AX998CC
2	Matt	0*F588886AX998CC

SQL db primary filegroup

Emp id	Emp_name	Emp_phone (varbinary(max)) FILESTREAM
1	Alan	0*E98886AX998CC
2	Matt	0*F588886AX998CC

Documents are stored in the file system and the db has a particular FILESTREAM

- Does not have to use high memory and memory buffer pool for caching Large objects.
- FILESTREAM enables caching at the system cache providing performance Benefits for large media files without affecting core sql server performance



SQL server db
Primary group

File system documents

Temp tables advantages

- Store data temporarily, large datasets needed to perform data transformation and modification

-- in-memory based optimized tables, schemas and data required to store until the db restarts.

-- required to store these tables in memory pools

-- retriction in terms of memory usage but storage for disk

Index Operators

OFFSET FETCH

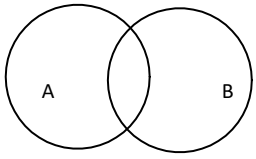
ID	Name
1	…
2	…
3	…
4	…
5	…
6	…

OFFSET clause - specifies the number of rows to skip before starting to return rows from the query.

FETCH clause - defines the number of rows to return after the OFFSET clause has been processed.

Skip first two rows and fetch next 4 rows only, we can use OFFSET and FETCH clauses with ORDER BY clause.

INNER JOIN -- helps to create a new table by combining rows which has matching values in two or more tables.



SQL Functions

11 September 2022 22:06

Functions

Special Functions in t-SQL
- Row Number Function
- Rank and Dense Rank Function
- Calculate Running Total in t-SQL
- NTILE Function
- Lead and Lag Functions
- FIRST VALUE Function
- Window Functions
- LAST VALUE Function
- PIVOT and UNPIVOT
- CHOOSE Function
- IIF Function
- EOMONTH Function
- DATEFROMPATS Function

Azure Cloud Fundamentals

08 September 2022 18:45

Data Warehouse Concepts

08 September 2022 18:46

Azure Fundamentals

08 September 2022 18:46

Basic PowerShell Scripting

08 September 2022 18:46

Apache Hadoop (Deep Dive)

08 September 2022 18:46

Azure Data Factory

08 September 2022 18:47

Azure Data Lake Gen2

08 September 2022 18:47

Azure SQL database

08 September 2022 18:47

Azure Blob Storage

08 September 2022 18:48

Azure Analysis Service

08 September 2022 18:48

Azure Synapse Analytics

08 September 2022 18:48

Apache Spark

08 September 2022 18:48

Python Programming

08 September 2022 18:48

Azure Databricks

08 September 2022 18:49

Overview of AWS

08 September 2022 18:49

Overview of GCP

08 September 2022 18:49