1. What is Microsoft Access?

Microsoft Access (or simply 'Access') is a database management system (DBMS). The program allows users to create databases and simple forms that serve as an interface to query the underlying data stored in the database. Users can display this data using these forms or create detailed reports containing information stored therein.

Access is a well-established software application featuring unique data storage formats with '.mdb' and '.accdb' file extensions. It is widely renowned as one of the most comprehensive and robust consumer database software applications. Like many other applications that are part of the Microsoft Office family, Access also features interoperability within the Office family. By that, we mean that the program can exchange and use data with other applications in the suite, such as Microsoft Excel, Microsoft Word, and Microsoft PowerPoint (Cox & Lambert 2013). In addition to the interoperability capability within the Microsoft Office ecosystem, the application also makes it possible to import data from and export data to other software applications and databases (Microsoft Corporation 2021a).

Despite its versatility and usefulness, some circumstances create the business need to move away from Access to a more encompassing and adequate solution. This need is usually rooted in positive business growth. Many businesses find that one of the most viable options is converting Access to SQL Server.

2. What is SQL Server?

Microsoft SQL Server (or SQL Server) is a relational database management system that has the fundamental function of storing and retrieving data for other software applications. These applications may be running on the same computer as the SQL Server installation or another networked system. There are several SQL Server editions for different customer segments depending on workload. These workloads range from single desktop application volume to enormous large-scale software system loads (Mistry & Misner 2014).

The descriptions above imply both the fundamental similarity and disparity between Access and SQL Server. While both are database programs, the former adequately serve most small organizations while the latter caters to medium-sized companies and large corporations. Additionally, while both products are from Microsoft, there are significant technical differences between the two on almost every front. These differences warrant specialized knowledge in both products. For instance, the keywords used in constructing queries in either program are different. Anyone wanting to convert Access to SQL Server must become familiar with the specific query syntaxes for both Access and SQL Server.

3. Why convert Access to SQL Server?

Access's maximum database size limit is 2GB, whereas the database size for SQL Server knows no upper bound. Access can handle a maximum number of 255 users at a time; contrastingly, the maximum number of concurrent users for SQL Server is unlimited. In terms of security, Access offers very rudimentary security measures that depend on the operating system's (Windows) permissions. On the other hand, SQL Server provides enterprise-level database security. For instance, unlike Access, SQL Server users cannot view and directly manipulate databases (Practical Computer Applications 2021).

On the subject of performance, Access's performance depends on the installation computer's performance or the network it serves. In comparison, SQL Server's performance depends on the installation server's very efficient performance (FMS 2021).

Regarding reliability, Access-based systems occasionally need database compact and repair operations. On the other hand, SQL Server features an automated database. Additionally, Access's reliability is generally dependent on the installation computer's reliability. In contrast, SQL Server-based systems rely on the installation server's reliability. And similar to the performance attribute, the server hardware is developed with reliability in mind given the workload that servers usually handle. Thus their reliability is nothing short of enterprise-level (FMS 2021).

SQL Server-based systems also feature transaction logging, which logs every change in data contained in the database. Therefore, prioritizing atomicity, consistency, isolation, and durability (ACID) attributes over hardware failures. This feature enables a database administrator (DBA) to review these logs for incomplete transactions and roll these incomplete processes back after a hardware crash. Also, the DBA can check and identify non-materialized but committed transactions and redo these transactions. These operations promote transaction atomicity and durability and help improve data integrity. On the other hand, Access lacks these features and completely loses out on this front (FMS 2021).

SQL Server also features table triggers, which is a mechanism for automatically executing a particular procedure when a pre-defined event occurs in a specific table or database. It is another feature used in maintaining data integrity. Access in comparison features limited support for these triggers. To be precise, '.mdb' databases do not support triggers. '.accdb' features macros that behave similarly to triggers but are nowhere near as comprehensive (FMS 2021).

Other unsung benefits attainable from converting Access to SQL Server include lower maintenance requirements and improved remote access (Practical Computer Applications 2021).

4. What is involved in converting Access to SQL Server?

Converting data from Access to SQL Server involves the following five steps:

1. Microsoft SQL Server Migration Assistant (SSMA) Execution.
2. Object Conversion.
3. Table Linking.
4. Testing and Revision.
5. Performance Optimization.

Access database objects can either exist in one or two databases, that is, frontend and backend databases—this split design aids network sharing. In a split design, the backend consists of tables and relationships, while the frontend comprises other elements such as queries, macros, forms, reports, VBA modules, and tables linking to the backend. Converting Access to SQL Server mirrors this setup. However, SQL Server now serves as the backend. This backend could be on an on-premises server or in the cloud. So, it is possible to retain the same frontend that users are used to while running SQL Server in the backend to leverage SQL Server's benefits (Microsoft Corporation 2021b).

Before embarking on the first step, it is advisable to involve an expert to complete some preparatory steps to ensure a smooth transition. These preliminary operations include adding table index and primary keys because every table in SQL Server must have a minimum of one index. And suppose any update operation is to ever occur in the table. In that case, any linked table must feature a primary key. Also, it is essential to verify primary and foreign key relationships by checking that they are associated with table fields featuring congruous data types and sizes. Finally, it is vital to assess the system in its entirety and manually remove attachment columns. This step is necessary because SSMA does not convert tables containing these attachment columns (Microsoft Corporation 2021b).

During the second step, the DBA executes SSMA, a tool that converts tables and parameterless Select queries. Other elements such as macros, forms, reports do not undergo a conversion. During conversion, SSMA initiates a DBA-driven object conversion by transforming Access objects into SQL Server objects and presenting the DBA with lists of converted Access and SQL Server objects. This presentation allows for a manual review of the conversion outcome. Also, it is noteworthy that the SSMA does not make copies of the converted elements yet. It is at this stage that the DBA confirms which objects be part of the conversion. The copying operation commences post-confirmation. The involved objects fall under the following categories, tables, and columns, parameterless Select queries, primary and foreign keys, Index and default values, and check constraints (Microsoft Corporation 2021b).

Following the transfer of Access's table, the DBA links each database table to its corresponding table in SQL Server in the third step. This mechanism enables the DBA to easily access the data contained in these tables without employing the more sophisticated SQL Server-specific database administration tools. However, since SQL Server now hosts the data, SQL Server (not Access) enforces the permissions applicable to the database(s) and associated tables (Microsoft Corporation 2021b).

In the fourth step, the DBA must remember that since only a subsection of queries goes through conversion, they must manually intervene to modify particular objects using T-SQL. Also, it is essential to note that there could be data types clashes between the two platforms. Some of the data types that often clash are large number data types, Boolean data types, date and time data types, and multi-value data types. So, rigorous testing after modifying any objects or revising any problematic data types is ideal at this point (Microsoft Corporation 2021b).

At the fifth step, the DBA should tune the new SQL Server database to improve database operations. To this end, the main cardinal recommendation is to ensure that small read-only queries occur on the client-side as this fosters access. The second recommendation is to execute long read-and-write queries on the backend to benefit from the server's comparative enormous processing power. Finally, the DBA should employ filters and leverage aggregation to mitigate traffic by reading and writing only the needed data (Microsoft Corporation 2021b).

5. How can [COMPANY NAME] help?

Successfully converting Access to SQL servers requires much more than theoretical knowledge. There are minute seemingly insignificant details that may hinder a seamless transfer. These details are both technical and operational. This part is where [COMPANY NAME] comes in. With almost two decades of experience in technology consultancy and software development backed by a multinational team of competent professionals. We could not be in a better position to help ensure that your business operations remain uninterrupted as we help you convert Access to SQL Server.

As a competent technology consultant, we understand that converting Access to SQL Server is usually involved. The degree of involvement is directly proportional to the project's size. So we make it a point to adopt an incremental approach while paying attention to detail. Our method ensures that we mitigate the risk of any failure associated with a sudden and hasty conversion attempt. Additionally, gradual migration makes it possible for us to identify process challenges and difficulties early that are unique to your organization. This knowledge allows us to recognize similar problems as we incrementally transfer you to SQL Server. This way, we continually improve the process as the project progresses.

Apart from the hands-on technical operations, our team also assumes auxiliary functions to ensure a smooth outcome. In particular, we meticulously document your current Access-based system, explicitly noting the critical areas that need refactoring. Additionally, we work hand in hand with the frontline Access users in your organization by setting up a system that enables iterative feedback that lasts throughout the project's lifetime. This system helps us identify and dynamically resolve the pain points that already existed in the Access-based system.

The extra work we invest in ensures that the new SQL Server-based system will be free from any functional issues that could frustrate users. This approach allows them to concentrate on getting used to the new processes to quickly bring their efficiency to optimal levels in the new SQL-Server-based environment.

6. Conclusion

Microsoft Access is a powerful database management system that has withstood the test of time. It provides a simple solution for small businesses, which makes it speedily adopted. However, in this simplicity lies its biggest flaw; size and capacity limitations. Progressive organizations that adopt Access sooner or later find their needs outgrowing the limits of the application. In our experience, this position prompts the need to convert from Access to a more sophisticated solution such as SQL Server. However, maintaining data integrity while seamlessly transferring to another platform necessitates safely converting Access's data to a format palatable by the target platform. This point is where [COMPANY NAME] steps in to make the complicated simple. [COMPANY NAME] continually assists businesses in developing and improving their IT infrastructure through innovative designs and solutions that align with their business objectives. And we take pride in developing pragmatic solutions that help businesses attain their business goals.

References

Cox, J. and Lambert, J. (2013) Microsoft Access 2013. PHI Learning Private Limited.

Introduction to importing, linking, and exporting data in Access - Access (2021a). Available at: https://support.microsoft.com/en-us/office/introduction-to-importing-linking-and-exporting-data-in-access-08422593-42dd-4e73-bdf1-4c21fc3aa1b0?ui=en-us&rs=en-us&ad=us (Accessed: 16 August 2021).

Migrate an Access database to SQL Server - Access (2021b). Available at: https://support.microsoft.com/en-us/office/migrate-an-access-database-to-sql-server-7bac0438-498a-4f53-b17b-cc22fc42c979 (Accessed: 16 August 2021).

Mistry, R. and Misner, S. (2014) Introducing Microsoft SQL Server 2014. Microsoft Press.

MS Access to .NET — Practical Computer Applications (2021). Available at: https://www.pcapps.com/application-services/conversions-and-migrations/ms-access-to-dot-net (Accessed: 17 August 2021).

When and How to Upsize Microsoft Access Databases to SQL Server (2021). Available at: https://www.fmsinc.com/microsoftaccess/SQLServerUpsizing/how/index.htm (Accessed: 19 August 2021).