Requisites for a Production SQL Server Administration Job?

SQL Server knowledge: (All my videos)

* Learn T-SQL
* SQL Server Administration Part 1 (SSMS)
* SQL Server Administration Part 2 (SSMS)
* SQL Server Administration Part 3 (SSMS)
* SQL Server Reporting Services Part 1 (SSRS)
* SQL Server Reporting Services Part 2 (SSRS)
* SQL Server Integration Services (SSIS)
* Basic Networking
* Books to read

What to do while applying for a job: Learn, Learn, Learn

Books to read and learn from:

<http://www.amazon.com/Professional-Microsoft-Server-2014-Administration/dp/1118859138/ref=sr_1_5?ie=UTF8&qid=1465095144&sr=8-5&keywords=sql+server+2016#reader_1118859138>

<http://www.amazon.com/Microsoft-SQL-Server-2014-Unleashed/dp/0672337290/ref=sr_1_fkmr3_2?ie=UTF8&qid=1465095298&sr=8-2-fkmr3&keywords=sql+server+ssrs+2014>

<http://www.barnesandnoble.com/w/professional-microsoft-sql-server-2016-reporting-services-and-mobile-dashboards-paul-turley/1123337547?ean=9781119258353&st=PLA&sid=BNB_DRS_Core+Shopping+Books_00000000&2sid=Google_&sourceId=PLGoP668&k_clickid=3x668>

<http://www.amazon.com/Professional-Microsoft-Integration-Services-Programmer/dp/1118850874/ref=sr_1_1?ie=UTF8&qid=1465095468&sr=8-1&keywords=sql+server+ssis#reader_1118850874>

<http://www.barnesandnoble.com/w/sql-server-2016-high-availability-unleashed-paul-bertucci/1123015496?ean=9780672337765&st=PLA&sid=BNB_DRS_Core+Shopping+Books_00000000&2sid=Google_&sourceId=PLGoP668&k_clickid=3x668>

<https://www.udemy.com/courses/search/?q=Raf%20Asghar&src=sac&kw=raf%20asghar&lang=en>

<https://www.dice.com/jobs?q=sql&l=CA&searchid=5301801110968>

<http://www.monster.com/jobs/search/?q=SQL-Database-Administrator&where=Los-Angeles-county__2C-CA>

<http://www.indeed.com/jobs?q=sql&l=California>

<https://www.google.com/?gws_rd=ssl#q=sql+dba+jobs>

Simple clear resume displaying your IT experience

Some experience in the following areas of Information Technology:

* SQL Server Administration
* Network Administration
* Any Programming Languages (T-SQL, C++, PHP, HTML, CSS, C, Java, JavaScript, Python, Shell, Ruby, C#)
* IT helpdesk

Posting on websites:

* Posting the Resume on the Web (Dice.com, Monster.com, Indeed.com, check google o for junior sql dba jobs)

Getting experience when you don’t have any (work for a relative, apply for intern with small companies)

What to expect if you get hired as a Junior SQL DBA

Junior dba responsibilities:

<https://datajobs.com/The-Cheesecake-Factory/Junior-Database-Administrator-Job~1739>

Prepare interview questions

**1. Explain about your SQL Server DBA Experience.**

* This is a generic question often asked by many interviewers. Explain what are the different SQL Server Versions you have worked on, what kind of administration of those instances has been done by you. Your role and responsibilities carried out in your earlier projects that would be of significance to the potential employer. This is the answer that lets the interviewer know how suitable are you for the position to which you are being interviewed.

**2. What are the different SQL Server Versions you have worked on?**

* The answer would be depending on the versions you have worked on, I would say I have experience working in SQL Server 7, SQL Server 2000, 2005 and 2008. If you have worked only the some version be honest in saying that, remember, no one would be working on all versions, it varies from individual to individual.

**3. What are the different types of Indexes available in SQL Server?**

* The simplest answer to this is “Clustered and Non-Clustered Indexes”. There are other types of Indexes what can be mentioned such as Unique, XML, Spatial and Filtered Indexes. More on these Indexes later.

**4. What is the difference between Clustered and Non-Clustered Index?**

* In a clustered index, the leaf level pages are the actual data pages of the table. When a clustered index is created on a table, the data pages are arranged accordingly based on the clustered index key. There can only be one Clustered index on a table.
* In a Non-Clustered index, the leaf level pages does not contain data pages instead it contains pointers to the data pages. There can multiple non-clustered indexes on a single table.

**5. What are the new features in SQL Server 2005 when compared to SQL Server 2000?**  
There are quite a lot of changes and enhancements in SQL Server 2005. Few of them are listed here :

* Database Partitioning
* Dynamic Management Views
* System Catalog Views
* Resource Database
* Database Snapshots
* SQL Server Integration Services
* Support for Analysis Services on a a Failover Cluster.
* Profiler being able to trace the MDX queries of the Analysis Server.
* Peer-toPeer Replication
* Database Mirroring

**6. What are the High-Availability solutions in SQL Server and differentiate them briefly.**

* Failover Clustering, Database Mirroring, Log Shipping and Replication are the High-Availability features available in SQL Server. I would recommend reading this blog of mine which explains the differences between these 4 features. [**Comparing the High Availability Features in SQL Server 2005**](http://learnsqlwithbru.com/2009/10/28/comparing-the-high-availability-features-in-sql-server-2005/)

**7. How do you troubleshoot errors in a SQL Server Agent Job?**

* Inside SSMS, in Object explorer under SQL Server Agent look for Job Activity Monitor. The job activity monitor displays the current status of all the jobs on the instance. Choose the particular job which failed, right click and choose view history from the drop down menu. The execution history of the job is displayed and you may choose the execution time (if the job failed multiple times during the same day). There would information such as the time it took to execute that Job and details about the error occurred.

**8. What is the default Port No on which SQL Server listens?**

* 1433

**9. How many files can a Database contain in SQL Server?How many types of data files exists in SQL Server? How many of those files can exist for a single database?**

* A Database can contain a maximum of 32,767 files.
* There are Primarily 2 types of data files Primary data file and Secondary data file(s)
* There can be only one Primary data file and multiple secondary data files as long as the total # of files is less than 32,767 files

**Added on Dec 30th 2010**

**10. What is DCL?**

* DCL stands for Data Control Language.

**11. What are the commands used in DCL?**

* GRANT, DENY and REVOKE.

**12. What is Fill Factor?**

* Fill Factor is a setting that is applicable to Indexes in SQL Server. The fill factor value determines how much data is written to an index page when it is created / rebuilt.

**13. What is the default fill factor value?**

* By default the fill factor value is set to 0.

**14. Where do you find the default Index fill factor and how to change it?**

* The easiest way to find and change the default fill factor value is from Management Studio, right-click the SQL Server and choose properties. In the Server Properties, choose Database Settings, you should see the default fill factor value in the top section. You can change to a desired value there and click OK to save the changes.
* The other option of viewing and changing this value is using **sp\_configure.**

**Added on Oct 29th 2011**

**15. What is a system database and what is a user database?**

* System databases are the default databases that are installed when the SQL Server is installed. Basically there are 4 system databases: Master, MSDB, TempDB and Model. It is highly recommended that these databases are not modified or altered for smooth functioning of the SQL System.
* A user database is a database that we create to store data and start working with  the data**.**

**16. What are the recovery models for a database?**

* There are 3 recovery models available for a database. Full, Bulk-Logged and Simple are the three recovery models available.

**17. What is the importance of a recovery model?**

* Primarily, recovery model is chosen keeping in view the amount of data loss one can afford to. If one expects to have minimal or no data loss, choosing the Full recovery model is a good choice. Depending on the recovery model of a database, the behavior of database log file changes. I would recommend you read more material on log backups and log file behavior and so on to understand in depth.

**Added on Nov 9th 2011**

**18. What is Replication?**

* Replication is a feature in SQL Server that helps us publish database objects and data and copy (replicate) it to one or more destinations. It is often considered as one of the High-Availability options. One of the advantages with Replication is that it can be configured on databases which are in simple recovery model.

**19. What the different types of Replication and why are they used?**

* There are basically 3 types of replication: Snapshot, Transactional and Merge Replication. The type of Replication you choose, depends on the requirements and/or the goals one is trying to achieve. For example Snapshot Replication is useful only when the data inside the tables does not change frequently and the amount of data is not too large, such as a monthly summary table or a product list table etc. Transactional Replication would useful when maintaining a copy of a transactional table such as sales order tables etc. Merge Replication is more useful in case of  remote / distributed systems where the data flow can be from multiple sites, for example sales done at a promotional events which might not be connected to the central servers always..

**20. What the different components in Replication and what is their use?**

* The 3 main components in Replication are Publisher, Distributor and Subscriber. Publisher is the data source of a publication. Distributor is responsible for distributing the database objects to one or more destinations. Subscriber is the destination where the publishers data is copied / replicated.

**21. What the different Topologies in which Replication can be configured?**

* Replication can be configured in any topology depending keeping in view of the complexity and the workload of the entire Replication. It can be any of the following:
* Publisher, Distributor and Subscriber on the same SQL Instance.
* Publisher and Distributor on the same SQL Instance and Subscriber on a separate Instance.
* Publisher, Distributor and Subscriber on individual SQL Instances.

**Added on Nov 12th 2011**

**22. If you are given access to a SQL Server, how do you find if the SQL Instance is a named instance or a default instance?**

* I would go to the SQL Server Configuration Manager. In the left pane of the tool, I would select SQL Server Services, the right side pane displays all of the SQL Server Services / components that are installed on that machine. If the Service is displayed as (MSSQLSERVER), then it indicates it is a default instance, else there will be the Instance name displayed.

**23. What are the different Authentication modes in SQL Server and how can you change authentication mode?**

* SQL Server has 2 Authentication modes; Windows Authentication and SQL Server and Windows Authentication mode also referred as Mixed Mode. To change the Authentication mode, read one of my blogs [**Changing SQL Server Authentication Mode.**](http://learnsqlwithbru.com/2010/07/17/changing-sql-server-authentication-mode/)

**The following Question and Answers on SQL Server High Availability were Added on Nov 28th 2011**

**24. What are the differences in Clustering in SQL Server 2005 and 2008 or 2008 R2?**

* On SQL Server 2005, installing SQL Server failover cluster is a single step process whereas on SQL Server 2008 or above it is a multi-step process. That is, in SQL Server 2005, the Installation process itself installs on all of the nodes (be it 2 nodes or 3 nodes). In 2008 or above this has changed, we would need to install separately on all the nodes. 2 times if it is a 2 node cluster or 3 times in a 3 node cluster and so on…

**25. What is meant by Active – Passive and Active – Active clustering setup?**

* An Active – Passive cluster is a failover cluster configured in a way that only one cluster node is active at any given time. The other node, called as Passive node is always online but in an idle condition, waiting for a failure of the Active Node, upon which the Passive Node takes over the SQL Server Services and this becomes the Active Node, the previous Active Node now being a Passive Node.
* An Active – Active cluster is a failover cluster configured in a way that both the cluster nodes are active at any given point of time. That is, one Instance of SQL Server is running on each of the nodes always; when one of the nodes has a failure, both the Instances run on the only one node until the failed node is brought up (after fixing the issue that caused the node failure). The instance is then failed over back to its designated node.

**26. List out some of the requirements to setup a SQL Server failover cluster.**

* Virtual network name for the SQL Server, Virtual IP address for SQL Server, IP addresses for the Public Network and Private Network(also referred as Hearbeat) for each node in the failover cluster, shared drives for SQL Server Data and Log files, Quorum Disk and MSDTC Disk.

**27. On a Windows Server 2003 Active – Passive failover cluster, how do you find the node which is active?**

* Using Cluster Administrator, connect to the cluster and select the SQL Server cluster.  Once you have selected the SQL Server group, in the right hand side of the console, the column **“Owner”** gives us the information of the node on which the SQL Server group is currently active.

**28. How do you open a Cluster Administrator?**

* From Start -> Run and type **CluAdmin** (case insensitive) and the Cluster Administrator console is displayed OR you can also go to Start -> All programs -> Administrative Tools -> Cluster Administrator.

**29. Due to some maintenance being done, the SQL Server on a failover cluster needs to be brought down. How do you bring the SQL Server down?**

* In the Cluster Administrator, rick click on the SQL Server Group and from the popup menu item choose **Take Offline**.

**The following Question and Answers were added recently. (on Dec 14th 2011)**

**30. What are the different ways you can create Databases in SQL Server?**

* T-SQL; Create Database command.
* Using Management Studio
* Restoring a database backup
* Copy Database wizard

**31. When setting Replication, can you have Distributor on SQL Server 2005, Publisher on SQL Server 2008?**

* No you cannot have a Distributor on a previous version than the Publisher.

**32. When setting Replication, is it possible to have a Publisher as 64 Bit SQL Server and Distributor or Subscribers as a 32 Bit SQL Server.**

* Yes it is possible to have various configurations in a Replication environment.

**33. What is the difference between dropping a database and taking a database offline?**

* Drop database deletes the database along with the physical files, it is not possible to bring back the database unless you have a backup of the database. When you take a database offline, you the database is not available for users, it is not deleted physically, it can be brought back online.

**34. Which autogrowth database setting is good?**

* Setting an autogrowth in multiples of MB is a better option than setting autogrowth in percentage (%).

**35. What are the different types of database compression introduced in SQL Server 2008?**

* Row compression and Page compression.

**36. What are the different types of Upgrades that can be performed in SQL Server?**

* In-place upgrade and Side-by-Side Upgrade.

**37. What is Transparent Data Encryption?**

* Introduced in SQL Server 2008 Transparent Data Encryption (TDE) is a mechanism through which you can protect the SQL Server Database files from unauthorized access through encryption. Also, TDE can protect the database backups of the instance on which TDE was setup.

**38. Does Transparent Data Encryption provide encryption when transmitting data across network?**

* No, Transparent Data Encryption (TDE) does not encrypt the data during transfer over a communication channel.

**39. What are the operating modes in which Database Mirroring runs?**

* Database Mirroring runs in 2 operating modes High-Safety Mode and High-Performance Mode.

**40. What is the difference between the 2 operating modes of Database Mirroring (mentioned in above answer)?**

* High-Safety Mode is to ensure that the Principal and Mirrored database are synchronized state, that is the transactions are committed at the same time on both servers to ensure consistency, but there is/might be a time lag.
* High-Performance Mode is to ensure that the Principal database run faster, by not waiting for the Mirrored database to commit the transactions. There is a slight chance of data loss and also the Mirrored database can be lagging behind (in terms being up to date with Principal database) if there is a heavy load on the Mirrored Server.
* There are very few things in the IT realm that cover as much territory as databases. Before the web, and more specifically before the advent of dynamic programming languages, these immense collections of data were the exclusive territory of the deep down programmers — rulers of their vast domain of cells and tables. As time progressed, however, more and more methods of accessing data and then displaying it in a way that was not hard coded became easier to understand and manipulate. Today, it is now possible for a user with no knowledge of database programming to create a database-driven website in under 3 minutes (I timed it), and manipulate it easier than it is to get around in the newer versions of a word processor that shall remain nameless. Therefore, the role of the Database Administrator (DBA) is one that truly has a huge variety of skill sets in it, especially since so many organizations have their own particular slants on things. Becoming a master in one particular kind of database does not necessarily mean that you could jump into another immediately without having to learn some of its own quirks. Therefore, we’ll try to keep this as neutral as possible — not going that deep into vendor-specific applications, but unfortunately this also means that we won’t be doing as deep a dive as we have in previous articles.
* That being said, we will be dividing up different questions you may face in a DBA Interview into 5 levels — least experienced to most experienced. That’s not to say that these questions cannot appear in different tiers — you very likely will see some of the level 1′s during a level 5 interview. Rather, this means that in order to reach level 5, you want to be comfortable with everything up to that point — not necessarily remember everything by rote, but at least be able to have a resource you can get the answers from quickly. So without further ado, let’s begin.
* 
* Level 1: The Minion
* Databases at their very core are methods of storing information for later use in a safe and secure place, using a strange language that at first it seems that everybody but you can understand. As a result, at this stage you most likely have a particular project that you need to maintain or implement — one that requires vast amounts of information to be cared for and kept tidy. Learning how to navigate a database isn’t necessarily as difficult as traversing an underground lair, but you can still get lost between the tables. Bananas are optional, but tasty.
* **What is a Database?**
* A Database Administrator needs to know what a database is before they can administer it, right? At its most basic, a database is a collection of tables, structured in such a way that it can be navigated like you would any sort of table. If you remember in math class, you may have had a number of tables that allowed you to quickly find a value if you multiplied an x and y value together — or in this case, what it would be if you were looking for a particular row and column value.
* **Why should I go to all the trouble of creating a database when I have a perfectly good Excel Spreadsheet?**
* Scale. If you were to take a (singular) spreadsheet and a (singular) table and place them side by side, there would be effectively no difference in the data you are seeing or what you could do with it. As you go bigger and bigger with more and more tables and spreadsheets, if you have a black belt in spreadsheet-fu you can accomplish many of the same tasks that a database could do as well. The problem is, as you go larger and larger and larger, that it becomes much more difficult to be human-friendly and still be efficient when it comes to processing data. So should you replace every single spreadsheet with a database? Not necessarily, but if the data on that spreadsheet needs to be accessed quickly by multiple users simultaneously and is growing rapidly, it may be time to consider going to the dark side (they have cookies).
* **What is a query?**
* A query in normal terms is a question, simple enough. It is the statement that is talking to the database in order to Create, Read, Update or Delete (CRUD) data. While many times a query is an actual question asking for an answer, it can also be the statement to modify, insert, or remove data in the database as well.
* **What is SQL?**
* Structured Query Language is the basic way of asking a database server to talk to you. Whether that is in the context of asking it a question, giving it answers to questions it is asking you, or updating answers that have already been stored in the database. The art of asking the right question is critical to getting back the right data you need, which is incredibly valuable when dealing with databases, as it is very easy to receive far more data than you know what to do with, or nothing at all.
* **What does ‘SELECT’ do?**
* SELECT in the terms of an SQL query triggers a question to the database. It looks across the specified table(s), finds the data you are looking for and then presents it to the user for consideration. Depending on the query, this can be an awful lot of data, so again, asking the right question is critical.
* **What is a primary key?**
* A primary key is usually used as the index for a particular table — a value that the table can depend upon to be a reliable unique value in every row. When trying to pull data for a particular row, the primary key will normally be used to pull that information, usually a numeric value. For example, if you are trying to pull up data on a specific person, and that database is using their unencrypted ssn as the primary key (naughty), then that could be used in the query to identify that particular row since there could be other people present in the database with that specific name or other identifying characteristics.
* **What is a Database Management System?**
* A Database Management System, or DBMS, is essentially the application that handles the heavy lifting between you (the user), and the raw data. The database itself is just that — the database; it cannot alter its own data any more than the average person can re-arrange their genetic code. The DBMS is what you are talking to when you are asking the questions. It is what looks at your question, thinks about it for a while, goes to the database, picks up the data, hands it back to you, and asks you to come again.
* **What is the difference between a Navigational database and a Relational database?**
* The best way to describe a Navigational DBMS is through that of a tree. Each value was associated with another through the use of a parent, most of the time with no other direct way to access the data. Relational Databases on the other hand use values common to multiple tables to establish a unique key — making sure that they are talking on the same page so that there are many, many ways to get to the same place. To put it another way, if you were trying to get from point A to point B, a navigational database would have one specific path to get there — via a freeway. A relational database on the other hand would have options for taking the freeway, a back road, a boat, a plane, a bus and sometimes a rocket — provided that each of those methods were set up correctly to talk to each other. Most modern databases use the relational database model.
* **Why do most database types not talk to each other?**
* In a word: money. In three words: a lotttta money. Different database vendors spend a huge amount of research time trying to find ways to give them a leg up on the competition; whether that may be by performance, storage capacity, longevity, reliability, scalability, the list goes on and on. As a result, trying to be compatible and backwards engineer every single feature of a particular database type is difficult in the extreme before you even get to the patent violations. Most databases can be simplified down to filetypes like .csv files, which can be used to transport basic data from vendor to vendor. That being said however, there would be a lot lost in translation without help from higher up.
* **What is a Frontend?**
* For those that don’t want to see row upon row upon row of data in front of them until they go cross-eyed, a frontend is essential. In essence a management program, a frontend allows admins to be able to view and modify high level database functions without the need to use the command line for every single thing. This can be extremely useful not only for efficiency, but also for safety, as it can prevent accidental data modification. It can also allow users that are more used to a GUI application most of the utility that the CLI permits.
* 
* Level 2: The Researcher
* You now can navigate a database backend at will, and depending on the rest of your skill sets, can access things that may or may not have you locked up in a federal penitentiary. However, when someone comes asking you for a question, you are the person they rely on for answers fast and accurately. You may end up with some odd overtime, but hey at least the medical’s good right?
* **What is a ‘join’?**
* Well when two tables love each other very much…not that much happens actually. However when you need to search across multiple tables simultaneously, a join can help make that happen. For example, if you were searching for information on a particular product and one table has the description while the other has pricing information, you can use a join to search across both tables simultaneously using a single query.
* **What is a foreign key?**
* When using a join or other type of query that goes across multiple tables, it can sometimes be difficult to make sure they are talking on the same page. A primary key can help with this, but sometimes this is impractical, and thus you need a secondary value that is consistent across multiple tables. For example, say that in a series of tables for product listings you have your primary key assigned to an auto-increment ID based on when the product was entered (a typical setup), and then none of these rows are able to line up with their counterparts in other tables. So if you have one table for product listings, another for price information, another for reviews, etc. — this could be a fairly major problem. However, if you know for a certainty that your part numbers for these products are going to be unique values, you can use that as a foreign key and suddenly everything lines up all nice and neat. This is possible since it exists in more than one table, and since is being referenced from outside its own table; it is designated ‘foreign’. This does not mean it still could not be the primary key for that particular table as well, it just means it has a reference that can be looked to from another point of view.
* **What is SQL Injection?**
* Also known as asking a question and getting the answer you want, rather than the answer they want to give you (anybody that has tried to navigate certain nameless support phones knows that this isn’t necessarily a bad thing); however in the context of a database application, this can be “a very bad thing”™. For instance, say that you are on an online banking website. You’re at the login screen, and it is waiting for you to enter your login and password so it can display your particular financial information. But what if you want to see the listing of everybody else that banks at this particular location? Depending on how the bank’s site is hardened against such an attack, you could get their personal information, current balances, PIN numbers, or even worse, enter your own data directly into the database — able to create new accounts, set up transaction history, active balances, the list goes on and on.
* **What is input sterilization?**
* One of the main answers to SQL Injection, input sterilization allows the database to selectively ignore data coming in from an input field and strip out non-required data. For example, if a field is expecting only a numeric value, there is no need for letters or symbols to be present in the user input. Therefore, these values can be safely ignored but still keep the functionality of the form intact. While not an end-all beat-all, it goes a long way to helping mitigate attacks on this vector.
* **SQL Vs NoSQL**
* NoSQL (Also called Not Only SQL), is a different form of database than the standard relational type. While it can use a lot of the same kinds of query language, it doesn’t necessarily use the same type of table structure that standard relational databases use, and thus in some cases can be more efficient. That efficiency depends greatly on its application however, and many times you will see NoSQL used in Big Data crunching and analysis applications that require real-time feedback.
* **What is ‘Big Data’?**
* DATA. If you’ve ever shopped on Amazon or at a Walmart, searched on Google or been on Facebook for more than 10 minutes, then you’ve seen Big Data in action. Big Data is essentially looking at the forest for the forest instead of the trees. An individual person is a unique entity with a specific set of actions and reasons for why they do what they do. Tracking an individual person’s actions can sometimes be useful, however it’s also a shot in the dark. But multiply that by many, many millions and suddenly the individual actions don’t matter as much — yet patterns start to emerge. A good example of this was published in the [New York Times](http://www.nytimes.com/2004/11/14/business/yourmoney/14wal.html?_r=0): Walmart discovered that just prior to a major storm, there was a run on the usual items such as bottled water, batteries and flashlights — but also strawberry pop tarts. This pattern was consistent across the board, so they were able to bundle these items together in certain parts of the store and increase profits. Amazon Suggestions, Google Analytics and other entities that run off of Big Data are huge moneymakers for their respective entities for being able to consistently give (relatively) accurate recommendations to users based on their past interests or purchases.
* **What is a ‘Flat File’?**
* A flatfile is a catch-all term used for concepts like Comma Separated Values (.csv). While there are a lot of different ways to create such a file, they all share ideas that they can be created and manipulated easily and without necessarily requiring a standard database application. These can also be used to transfer data from system to system due to their lightweight status. In some cases, these have been replaced by XML files, however XML can when compared to certain kinds of flatfiles, be very large.
* **I have a database that was built in MySQL, and I need the data to be moved over to Microsoft SQL Server. How would I do this?**
* The easy answer would be to contact Microsoft Tech Support and bring your checkbook. A more difficult answer would be to bring it down to a .csv file and then import it into SQL Server, but without a specialty conversion utility you may lose some program-specific specific tricks, thus requiring some rebuilding once the conversion is complete. This is not saying that this would work in all cases, but it is at least an option.
* **What is the difference between ‘=’ and ‘LIKE’?**
* When crafting a query, or using programming to display data in certain ways depending on the values being returned, you may want to think that these can be used interchangeably. There is one big difference, however: equal means equal. The value being returned must match the value it is being compared to 100%. LIKE, however, can be used with a number of different wildcard mechanics, allowing you to be a bit more flexible in your rules.
* **What is a Null Value?**
* A Null Value is an absence of data. This one is a bit misleading sometimes, because depending on who you ask, it can be considered many possible things. “Null equals 0”- Not in this context, because 0 is a value. “Null equals Empty” — closer, but again sometimes an empty value can still be considered a value depending on how the field is structured. If a column allows for null values, and no value is submitted, then it allows it to be Null.
* 
* Level 3: The Riddler
* “Riddle me this, riddle me that.” The ability to ask the right question can unlock as much information as you can handle, and in some cases more than you ever want to know. However, just asking questions isn’t enough, since in order to have questions answered, there must be data present in the first place. Being able to generate data and then know that it is safe for future use is vital, since you never know when some scrap of information might be useful in your next ~~caper~~ project.
* **What does ‘INSERT’ do?**
* INSERT submits data into a database as a new row, usually through the use of a form. While forms can take many…forms…, the most common uses are through either a dedicated application or through the use of an HTML form. Clicking on the ‘submit’ button will trigger the built in form reaction to scan the form for particular fields, making sure the required ones are entered correctly, make sure the user isn’t being naughty in what they are trying to enter, then submit the data to the database.
* **What does ‘DROP’ do?**
* DROP removes a table from a database or a database from a server. A very dangerous command indeed, it is only to be used in situations that absolutely require it, as unless you have a backup of it handy, there is no coming back from this.
* **What is the difference between T-SQL and PL/SQL?**
* T-SQL or Transact-SQL is Microsoft’s version of SQL. The main additions Microsoft made to the main branch of SQL involve the addition of procedures or routines — scripts essentially — that can be run under certain criteria. PL/SQL, on the other hand, is Oracle’s version of SQL, and conceptually the two are very similar. However, because of the nature in how they were developed, trying to move data from one to the other involves quite a bit of work. The main differences deal with how they multi-task and how they lock elements when they are in use.
* **What does ‘UPDATE’ do?**
* UPDATE allows values to be modified where they meet specific criteria. For example, say that you were on Amazon and were about to move. As a result, you would want to adjust your mailing address so that you actually got your stuff. You would therefore go into your settings and it would show you your current address. Modifying this address and then submitting the form would update your address based on your particular user profile. If it updated anybody else’s address to match that would be a serious problem — at least for the person doing the paying.
* **Why do database servers benefit from a lot of memory, and why do 64-bit operating systems help in this regard?**
* Database servers like to cache as much data as possible when they are reading it a lot. Storing this information in active memory is a lot faster than trying to find it again from the hard disk or other media. Therefore more memory = faster response time = better performance. The problem is that for most operating systems the maximum amount of memory that can be used by a 32-bit OS is 4 gigabytes. While in years past this would have been an inconceivable number, today it is a drop in the bucket. 64-bit operating systems resolve this issue by being able to handle memory to 192 gigabytes currently for Windows, while Linux can theoretically go much higher at present, and these numbers will only climb higher and higher.
* **Why is it a bad idea to run a test on a live database?**
* On a test database, it’s relatively easy to keep the performance variables to a minimum. On a live database however, it needs to be functioning for all users all the time. Running untested code on a production database can not only reduce performance, but also create unforeseen instability in the server itself — potentially causing crashes and data corruption.
* **Why is it difficult to use standard file by file backup methods on an active database server?**
* This problem is twofold. First, many database servers place locks on database files that are currently in use. Most backup programs that try to do a file-by-file backup will therefore be unable to create a copy of this file, as they cannot get exclusive permissions to it. Second, while some database servers have only a single file to backup a database, others have multiple files that can be stored in different locations across possibly multiple physical hard disks. The problem can be resolved in one of two potential ways. First, using the backup method within the database server itself. Some programs such as Microsoft SQL Server allow you to create a scheduled backup directly within the server application to a location of your choosing. Others require you to use a scheduled task or another on-demand type of backup solution. The second would be to use a backup application that can talk directly to the database server, allowing the database to be backed up using a different technique.
* **When would you use an offline backup method versus an online backup?**
* If the above methods are unavailable when trying to create a backup solution, another potential method is temporarily taking down the database or database server in order to create a file-by-file backup. The problem with this method is that if the server goes down incorrectly, the backups could be flagged as bad and thus unusable. **Periodically testing your backups to make sure they are working properly is strongly recommended, regardless of what method you use to create them.**
* **What is Replication?**
* Database replication allows for real-time automated backups between multiple database servers. This allows for the creation of either a fall-over server, or warm backup for use in case the main server goes down.
* **Is data in databases encrypted by default?**
* While most database servers support some form of encryption out of the box, it is not enabled by default due to performance hits and security concerns.
* 
* Level 4: The Librarian
* Data Organization is critical to being able to navigate through large numbers of tables and much larger quantities of data. Being able to look ahead to potential applications and create and optimize tables in a way that allows them to grow without having to rebuild them entirely can help save many hours of hassle. Bananas are still optional and still tasty.
* **What is the difference between a ‘TINYINT’, an ‘INT’ and a ‘BIGINT’?**
* Contrary to popular belief, this is not the lifecycle of a talking tree. Rather, it is the creation of a column that allows for specific levels of integers (numeric whole numbers) up to a specified cap. There are many ways to limit the growth of fields, but in the case of [Microsoft SQL Server, these each represent a value in bytes](http://msdn.microsoft.com/en-us/library/ms187745.aspx), which creates a maximum value that the field can hold.

|  |  |  |
| --- | --- | --- |
| Data type | Range | Storage |
| bigint | -2^63 (-9,223,372,036,854,775,808) to 2^63-1 (9,223,372,036,854,775,807) | 8 Bytes |
| int | -2^31 (-2,147,483,648) to 2^31-1 (2,147,483,647) | 4 Bytes |
| tinyint | 0 to 255 | 1 Byte |

* **How would you store files within a database?**
* Two common ways to store files for use by a database are either within the operating system’s file system, or within a field of the table itself. Uploading and storing the files outside of the database makes for faster creation of the application, and can be more efficient if the file sizes are larger, but can potentially cause security issues if the files are not secured correctly. On the other hand, the files can also be stored directly within the database using a BLOB-type field. A BLOB is a Binary Large Object, essentially an empty area where a file can be uploaded to but not exceed a specified limit. Like int in the example above, blob has a number of different potential sizes, depending on the type used. Bear in mind there are other methods for storing and accessing files in a database server, these two are merely the most common.
* **When would you use ‘char’ versus ‘varchar’?**
* This is a bit of a difficult question, mostly because it depends so much on what your application is. For example, if you have a form field that can be nearly any length and changes every single time, then varchar is a much more practical choice, since it gives you much more flexibility. If however you have a field where every value is going to be exactly the same length, then you can get more efficient performance out of a char. Again, it depends on exactly what your application is, and how you plan to cook it — seasoning as you see fit.
* **What is XML?**
* Extensible Markup Language (XML) is a fast way to display data that not only conforms to a structure that can be read by machines, but is also easily understandable by humans. Because they can be dynamically and manually generated in many different ways, they are easy to produce and map to; and because they retain the same structure despite the data being updated, they can be relied upon for automatic functions such as RSS aggregation.
* **What shows that a database server is running?**
* <Insert joke about needing to catch it here/> Database servers run as services or daemons, most times in the background without the necessity to see that they are running in order to interact with them. When things go sideways however, being able to verify that the service is in fact up and running can be an excellent place to start troubleshooting. Checking under the services area of your particular operating system, whether that be by GUI or by CLI, can show you that the service is started or not, thus allowing you either to start or restart it as need be.
* **What is WYSIWYG?**
* What You See Is What You Get. A mouthful of an acronym, it allows for the creation of an application that is consistent regardless of how it is viewed — whether on the design screen, being viewed in a browser or being printed. Creating an interface to a database that is not only functional but also looks nice is a trick in itself, and can take a lot of work to get it just right.
* **Why is it frowned upon to use ‘SELECT \* ..’ in a large database?**
* Picture it like a group of people in line for a bathroom, and every single person that was going in there was going to use the toilet, change their clothes, take a shower, iron their jacket, take another shower, etc. There is only so much area that can be used efficiently before you start to get a queue, slowing down the whole operation that can eventually cause the entire thing to collapse under lack of toilet paper. You can quickly get back more than you can use or understand, so optimization is key when creating queries and asking only what you need to get the question answered.
* **How would you get the quantity of results from a query?**
* COUNT() is the main supported way to be able to get the number of returned results from a query. While there are many other options such as mysql\_num\_rows, these are considered obsolete and are being removed.
* **What is a Database Schema?**
* If you’ve ever seen one of those Visio diagrams with 40 different tables with lines connecting particular columns on one with those on another, that’s a database schema. Essentially a two-dimensional representation of how each table talks to other ones, it is the way to view the design of a database as a single entity and not as a jumble of different tables.
* **What are Nested Queries?**
* A query within a query, this particular method can be tremendously difficult to troubleshoot and even harder to manage without a lot of overhead. In most cases, a nested query can be replaced with a JOIN, allowing for much more efficient use of resources.
* 
* Level 5: The Caretaker
* You understand the system inside and out and protect it as such. If anything happens, you’re the first one to pick up on how bad it can get and the last one to go home after its been fixed. Sometimes pictured as a miracle worker, it’s your job to make sure the project keeps going, building a team that can find new ways to improve upon the designs, and keep the wheels spinning.
* **What is ODBC?**
* Open Database Connectivity is a way to make different kinds of frontends talk to different data sources (DSNs) such as Databases. The specifics available depend on the type of application being used, the driver being used and the backend to which it is being applied.
* **For Oracle systems, what is OFA?**
* Optimal Flexible Architecture (OFA) is the recommended layout for installing and configuring an Oracle database.
* **For Oracle systems, what is error “ORA-01034”?**
* The full error is “ORA-01034: ORACLE not available”. While there are many potential causes, the most common is that the service is just not running. The resolution is to start the service, then see if the error comes back.
* **What is Normalization?**
* When most people first start working with databases, the first instinct is to create massive tables for storing data — one place, one query — keeps things simple. However, as they grow to unmanageable levels, it is a good idea to look into Database Normalization. This idea allows for data to be split off into smaller more efficient tables that (hopefully) reduce the amount of duplicate data. In this way, smaller queries can be run on individual tables instead of having everybody always talking to one big one — thus improving performance.
* **For Microsoft SQL Server, what is a DMV?**
* Dynamic Management Views are functions built into Microsoft SQL Server that allow for troubleshooting, diagnostics and server health monitoring.
* What are the default ports for MySQL, SQL Server and Oracle, and can/should this be changed?
* The default port for MySQL is 3306, and can be changed in Windows as noted in [this article](https://dev.mysql.com/doc/refman/5.1/en/mysql-config-wizard-networking.html) or in \*nix as noted in [this article](http://askubuntu.com/questions/407847/how-to-change-mysql-port-number-in-ubuntu). The default port for Microsoft SQL Server is 1433, and can be changed as noted in [this article](https://support.microsoft.com/kb/823938). The default port for Oracle is 1521, and can be changed as noted in [this article](http://library.blackboard.com/ref/df5b20ed-ce8d-4428-a595-a0091b23dda3/Content/_admin_server_database/database_oracle_changing_database_ports.htm). Depending on your security stance, changing the port that your database server uses can be a good way to lower your profile and reduce the amount of unauthorized access attempts against the server.
* **For Microsoft SQL Server, What is Log Shipping?**
* A form of backup on Microsoft SQL Server, Log Shipping is similar to replication and allows for rapid failover if the main server goes down. One thing to bear in mind, however, is that a log shipping based failover must be activated manually; it will not switch over automatically.
* **For Microsoft SQL Server, what is DBCC?**
* Database Console Commands (DBCC) are a series of utilities for SQL Server designed for maintenance and reporting. A full list of the commands can be found [here](http://msdn.microsoft.com/en-us/library/ms188796.aspx).
* **What is Cloud Computing?**
* Cloud Computing is usually a catch all term for data being stored “over there”. Placing high-requirement applications onto dedicated hosting services can be beneficial depending on the application, however it can also cause catastrophic security problems and availability issues. It is therefore highly recommended to keep important data in-house, and only outsource in situations that it cannot be avoided. Cloud Computing, Big Data and Data Mining are many times talked about in the same sentence since processing power required for one usually means the others become viable either as a requirement or a side effect.
* **What is Hadoop?**
* Hadoop is a Data Mining application designed to handle very, very large amounts of data across a wide variety of environments — from one to thousands of systems. Used in situations that don’t necessarily fit into standard database structures, its main strength is being able to take one giant project and split it off to each of its member servers, have them each process their own job, then have their findings recombined into one viewable result.

**what to expect first day?**

installing and configuring sql server

creating database and tables

creating some scripts

backup and restore

security

movement of data

creating reports

check the status of performance

**what not to expect:**

creating HA solutions

clustering

application performance issues

The start of a new year is a time for self reflection and along with it perhaps a desire to do something new with your career. Maybe you are considering becoming a Junior Database Administrator (DBA)?

Acquiring your very first role as a Junior DBA is not easy. Just browse any one of the many job sites advertising SQL Server DBA job vacancies you will quickly come to realise that almost all Junior positions list a number of years of SQL Server experience as a necessary requirement.

**How Most Junior DBA’s Earn Their Stripes**

The truth is that a significant proportion of SQL Server Database Administrators (DBA’s) start out in an alternative discipline, that is to say they are Accidental DBA’s. For example, a Windows ServerAdministrator may also be tasked with looking after the businesses SQL Server infrastructure, perhaps as the result of a DBA having left the company or through cross training opportunities within the organisation. These individuals acquire the necessary experience on the job within their existing role in order to make the transition to a Junior DBA position.

If you are seriously interested in either starting out on the path to becoming a SQL Server Database Administrator or perhaps wish to make a lateral move from an alternative IT discipline, this revelation can be quite disheartening. You may feel that it is somewhat of no win scenario, as right now you have limited experience working with SQL Server, hence why you are considering junior opportunities in the first place.

**Help is at Hand**

Well fear not my friend, for all is not lost. It may surprise you to hear that I am of the opinion that SQL Server experience alone is NOT the most desirable quality of a Junior DBA! You see the Accidental DBA will often be self taught and so although more experienced than perhaps you are currently, they have acquired their knowledge of SQL Server the hard way. The very same knowledge can be passed on to a newly hired Junior DBA by an experienced mentor in a fraction of the time. For this reason there are a number of very desirable qualities other than SQL Server experience alone, that can provide you with a serious edge over the competition.

Over the years I have interviewed many aspiring data professionals and I want to share with you what I look for when hiring Junior DBA’s.

**My Top 10 Interview Tips for Junior DBA’s**

**1. Are You a Problem Solver?**

At the core of being a successful Database Administrator is an ability and a drive to problem solve.

You need to be able to effectively demonstrate your passion for problem solving. Problem solving is not really a skill that you can teach but it can certainly be improved and enhanced through practice.

You must be able to demonstrate a good level of problem solving to the interviewer.

Be prepared for your interview whether it be face to face or over the telephone, with a whole host of examples that demonstrate your ability and flair to problem solve.

**2. Do You Know The Primary Responsibility of a DBA?**

Do you know what the [primary responsibility of a DBA](http://www.johnsansom.com/the-database-administrators-primary-responsibility/) is? If not then you need to read this now! Think of it as the DBA Prime Directive. Everything that you do as a data professional is built on this.

**3. What do You Know About Database Backups?**

Ideally you should get to grips with the basics of SQL Server backups but as a very minimum you must know why they are necessary and why they are important.

**4. Why do You Want to be a DBA?**

I look for candidates who know why they want to be a Database Administrator. There is no right or wrong answer here but you must be able to demonstrate to the interviewer that your reasons are clear and have been thought out.

**5. Can you provide quality examples of your previous projects?**

You will almost certainly be quizzed by your interviewer on the details of your past projects. This will enable you to demonstrate a wealth of qualities about yourself.

The key is to ensure that you have prepared ahead of time, a number of projects that you can discuss with your interviewer. Although preferable, your examples do not have to be SQL Server Specific or even IT Project work for that matter. Anything from University projects to work experience or examples of problem solving from part time jobs. Just be sure to have sufficient detail readily available and clear in your mind.

This is your opportunity to demonstrate to the the interviewer any number of your qualities and skills. To get you started, you may with to consider how you can discuss your project work to demonstrate:

* Working in groups
* Working independently and without supervision
* Planning
* Project Management
* Time Management
* Problem Solving

**6. Can you Demonstrate Overcoming Adversity?**

Be prepared to provide examples of how you have overcome problems or difficulties.

Show that you are keen to develop professionally by providing insight and evaluation of your own performance. Consider, ahead of time, how could you have done things differently?

**7. Do You Have a Basic Grasp of the Relational Model?**

You must at least learn and understand the fundamentals of [The Relational Model](http://en.wikipedia.org/wiki/Relational_model) before applying for a DBA position. There is simply no excuse in the Internet era for not having an overall understanding of the basics. Don’t like studying online then get yourself a good book. Perhaps [Database Systems](http://www.amazon.com/exec/obidos/ASIN/0321210255/josasqsedbint-20) by Connolly & Beg would be a good place to start for you and is an excellent text used by many Computer Science Degree courses.

**8. Do you Have a Personal Development Plan?**

In order to be a data professional you have to continually be learning about your chosen subject. In an ideal world your employer would assist you with this endeavour however you should take charge of managing your own professional development. Your prospective employer may have budget to allocate to your training but may not know how it should be spent or what courses would be appropriate for you. Individuals that can clearly demonstrate that they are proactive about managing their own professional development are sought after.

Consider how you are going to increase your skills and knowledge of SQL Server. The more detailed your plans the more attractive a proposition you are to a perspective employer.

Some tips to get you started include:

* Join your local SQL Server User Group
* Study for SQL Server Certifications such as MCTS or MCITP.
* What SQL Server texts are you currently reading and would like to read? You have a reading list right?
* Follow SQL Server Bloggers by subscribing to their RSS feeds.
* Follow SQL Server DBA’s on Twitter to keep up to date on what’s going on the SQL Server Community.
* Visit SQL Server Forums to both review questions and contribute answers.
* Frequent some of the many excellent SQL Server Community sites such as [SQLServerCentral](http://www.sqlservercentral.com/), [SQLServerPedia](http://www.sqlserverpedia.com/) and [SQLTeam](http://www.sqlteam.com/).

**9. Do You Have Any Goals?**

Ensure that you can provide details of both your short (within the next year) and medium term (next five years) goals. This may link in to your Professional Development plan but you may also wish to include details of your life goals. Again there is no right or wrong answer here, as the objective is to demonstrate that you are forward thinking and ambitious but if you want to get serious about your goals then you will want to make sure that they are [SMART](http://en.wikipedia.org/wiki/SMART_criteria) goals.

**10. Do You Know What You Don’t Know?**

The more you learn and the more experience you acquire, the more you will realise what you don’t know. Let me tell you now that there is not a single data professional that knows absolutely all there is to know about SQL Server. It is simply too large a subject. That is why there are specialist SQL Server roles, such as Database Developer, Database Administrator, Business Intelligence Analyst, Report Writer and many more.

The key point to take away here is that it is perfectly acceptable and expected for you to not know the answer to a particular SQL Server question or problem. Personally I consider it a strength to know what it is that you do not know. Knowing your limits means that you can identify when it is time to call in the cavalry.

When you are stumped by a question in your interview don’t panic! Simply tell the interviewer that this is not an aspect of SQL Server that you are currently familiar with and then immediately go on to explain, if possible, how you would go about finding out the information required. Remember, it’s all about problem solving. There will be times in you career as a data professional that you do not have the answer to an issue immediately before you but if you have the means to find the solution then you are indeed a valuable asset.

**What Are Your DBA Interview Tips?**

I’d love to know what are your DBA interview tips? Leave your thoughts and feedback below.

Filed Under: [Interviews](http://www.johnsansom.com/category/professional-development/interviews/), [SQLServerCentral Syndication](http://www.johnsansom.com/category/ssc-syndication/), [SQLServerPedia Syndication](http://www.johnsansom.com/category/sqlserverpedia-syndication/) Tagged With: [Interview Tips](http://www.johnsansom.com/tag/interview-tips/), [Junior DBA](http://www.johnsansom.com/tag/junior-dba/)