Recruiting Fundamentals Training

Database Systems
An Overview

IT 4 Recruiters

IT4Recruiters.com Confidential

Revised: Rob Broadhead

IT4Recruiters.com
Suite 300 #288
115 Penn Warren Dr
Brentwood, TN 37027

Contact Name: Rob Broadhead

Printed in the United States

Overview

The concept of a database has been around since the late 1960's and it has evolved substantially since those early days. The Internet dictionary defines a database as: "a comprehensive collection of related data organized for convenient access, generally in a computer." This definition is a good one outside of the IT realm, but there is much more detail required when discussing database resources or projects in IT. This document will look at the state of the database in the modern IT world and the various types of databases that are in use. The big vendors in the industry will be discussed, as well as the many types of databases. This will all paint a picture of the way data is stored and manipulated in modern applications.

Keywords

Here are some useful definitions to keep in mind while reading this document:

- SQL: (Structured Query Language) A standard language for making requests of a database. Also called "Sequel."
- RDBMS: (Relational Database Management System) The main engine and tools for storing data based on relationships.
- OODBMS: (Object Oriented Database Management System) The main engine and tools for storing data based on objects they belong to.
- NoSQL: Not Only SQL, this refers to database engines that have a higher level approach than SQL to access and manipulate data, often decreasing ramp up time in using the database.
- ISAM: (Indexed Sequential Access Method) A method of storing data in files where each row is a record and an index file can be paired with it to improve search times.
- Database Application: Complete solutions often aimed at non technical users including Access, QuickBase, FileMaker/Bento, etc.
- Embedded Database: An approach to storing data where the data is local to the device and does not require a connection to a server.
- Aggregation: Combining a set of data. This might include generating a total number of sales for a month, generating the average employee age, or a number of other ways to slice and dice a set of data.

Databases Snapshot

The database landscape is in a state of flux as mobile and "cloud" applications become more prevalent. These new application approaches have new requirements and spawn new solutions for storing, accessing, and manipulating data. There is also a substantial amount of growth in the database industry driven by the larger amounts and varied types of data. The faster network access that is widely available has also pushed the advance of databases along with a greater amount of data consumption in general.

History

The need to store data is as old as computers and programs. Even a simple address book application is much more useful when a database is attached. The database started as something similar to a modern text file and has grown into computer data structures and engines as complicated as anything done with computers today. Data is the lifeblood of applications and improved ways to store and retrieve data lead to improved applications overall.

Databases of some sort have existed as long as programs. An app. often needs a place to store data.

The idea of a database progressed from storing data as lines in a file...

record1 Tim Tester 123 Main Street
record2 Sam Sample 22 Baker Street
record3 Dave Demo 1600 PE avenue

...to indexed files paired with a text file, to complex data storage mechanisms paired with equally complex indexing structures. Early forms of database models are still in use, including the very popular comma separated values (CSV) that is used to import and export data in a lot of applications. Fixed width files are common in financial applications and are faster to process, but less flexible than CSV formats. Fixed width storage allows data to be accessed by row-column coordinate pairs, while CSV requires row and an Nth column after parsing the row into columns.

Databases have progressed from simple (flat) files to fixed width, delimited (CSV) and beyond.

Note that the early forms of databases were essentially a grid structure and were easy to understand. This same structure spilled over into spreadsheets and even today the same approach is presented to users of spreadsheet applications. Spreadsheets

Spreadsheets (grids) are an easy to read form of a database and often used for maintaining data.

have grown to support more than a simple, two-dimensional model by the use of tabs or pages to provide a third dimension to data.

Data has gotten more complicated in step with the applications that use the data. The amount of data consumed has also grown substantially as memory, storage, and processing power has been able to remove limits. Thousands of rows of data were a sizable program in the 1960's and now small programs handle millions of records. How many email records do you think are stored in your personal email client application?

Data has grown from thousands of records to billions in the last few decades.

The easiest visual example of storage and speed of data is in computer animation. There is a lot of computation that goes on related to animation, but first, everything that is animated has to be described by the data. A scene from Disney's Toy Story may only have a couple of characters on the screen, but the data to render the screen includes all of the points required to draw the characters. It also includes all of the data to describe colors and lighting for the scene, and all the background objects and colors. The data for the background alone may be huge in cases where the background is more than a simple backdrop. In Disney's Monsters Inc. Sully, one of the main characters, (a big, blue hairy creature) was designed with the data for all the hair on his body stored. This data was then used to calculate a realistic look as the fur/hair responded to movement and wind.

According to Wikipedia the Sully character in Monsters Inc. has 2,320,413 strands of hair. That is just one character definition that is stored in a database.

Databases in Action

Storing data is only the beginning of why a database is used. A database also provides a way to quickly retrieve data (search for needles in a haystack) and group data. Online banking may seem like a straightforward data task, but think of the speed at which a typical banking web site can return all of your transactions based on how you want to look at them (date, check number ranges, etc). In particular, realize that those are just a few of the data records in a system of potentially millions and even billions of transactions. Nearly every search made on a daily basis is searching for a needle (or group of needles) in a haystack and results are often returned in sub-second times.

You interact with databases every day including: email, search engines, address books, and calendar programs.

The fast retrieval times are accomplished through optimizing how data is stored and through the use of an index. An index is a quick way to jump to a group of data just like an index in a book. In the case of data the index may point to a “page” of memory where the data is stored or may be more complex and be a short cut to the

An *Index* is used to help a program access data quickly.

exact location of the indexed value. In the example below we see a very simple index method. Consider the first column to be an index on the first name column (column 2). In a search each record could be reviewed or the index could be used to jump to what is searched for. As an example, a search for someone named Curtis could use the index to jump to the "C's" in the data file and search until the "C" records were all reviewed. In the example this would mean checking only one row instead of 5.

A Simple Index example:

A	Andrew Allen	123 Main Street
A	Allen Jones	22 Baker Street
B	Bob Nobody	111 Second Ave.
B	Bill Somebody	33 First Street
C	Charles Anybody	444 Broadway

An index allows a value to be looked up without looking at all of the other values. This can reduce queries from hours to seconds.

Data and the ability to index it gets more complicated than this, but the important thing to know is that proper index usage can reduce queries from hours to seconds.

It is easy to see from the importance of databases that money can be made from the design and creation of a database system. There are smaller players in the industry, and even start-ups, but the big names in databases are tied to the big names in IT. For example, Oracle was a database company from the start, IBM has always had database offerings, and Microsoft became more of a database player as they grew from personal PCs to a major Business player. Even the new kids on the block: Amazon and Google, have database related offerings as they try to cut into the market held by the other IT giants. The solutions available from all of these players also have open source offshoots and less featured "personal" or "developer" solutions aimed at bringing as many developers as possible into the fold.

Many database vendors have an open source or free version called lite, personal, or developer.

All of the big players have whole sections of their web sites devoted to all things database and some level of developer community around the technology. The industry has grown to the point that professionals can literally make a career out of working in (and on) a single database system, and they can be rewarded handsomely in a financial sense.

Databases are a big enough part of IT that careers can be based on a single vendor.

These databases all have their own unique approach to solving data storage and retrieval tasks, but they do tend to share a

common language. Structured Query Language (SQL) is the universal language for working in most database systems. SQL gives a baseline language that can be used from database to database and makes common tasks such as inserting, deleting, updating, and retrieving data skills that are portable from one database to the next. IT being what it is, each of the database implementers also have their own extensions to SQL to do additional data work. This includes work such as date manipulations (find all weekdays in a date range), output (display a result to the console), file handling (save output to a file in xml format), and a wide range of other useful features. The larger database engines including Oracle, Microsoft, and IBM DB2 have a huge list of extensions that make database queries as complex as any programming language solution. The smaller database engines have less complete (and less complex) proprietary languages, but this does not mean experience in a specific database query language will help one to do things faster and easier than someone using only standard SQL. As an aid in distinguishing some of these extension languages see the table below:

SQL is the language used by most databases, but each vendor has their own extensions.

Database Engine	SQL based query language
Oracle	PL/SQL
Microsoft SQL Server	T-SQL
IBM DB2	SQL PL
MySQL	SQL/PSM
Interbase/Firebird	PSQL
PostgesQL	pgSQL

The vendor specific SQL version is often the key indicator as to which DB is being used by a customer.

These languages are often the way to create and maintain complex database objects including: triggers, stored procedures, functions, and the previously mentioned indices. These are more advanced database features but I think it is safe for our purposes to provide a quick description of each.

Advanced DB skills include: stored procedures, triggers, index creation, and functions.

- Trigger: A database trigger is used to make a change or perform a computation when data is added, deleted, or modified. For example, set the date_changed column to the current time when the row data has been changed.

Triggers perform an action when data is changed.

- Stored Procedures: A way to expand standard queries by including looping structures, computations that link queries, and many other approaches that bring complex logic into the ability to query. For example, a stored procedure might be called to query a series of tables, find the changes, and log them to a file.

Stored Procedures are often complicated queries and and use advanced database logic.

- Functions: These are very similar to Stored Procedures and typically take inputs, do some work, and provide an output. A simple example would be a function that counts the number of empty columns in a query result.

Functions are similar to stored procedures and require similar DB skills.

The definitions may make these features seem simple and easy to understand, but having experience creating these is often a way to distinguish someone that has used a database vs one that has truly built or maintained a database.

Database Approaches

The way data is stored and retrieved can be varied from database engine to database engine. Although every vendor has their own detailed implementation, there are some high level approaches to how databases work that is a distinguishing factor when looking for candidates. These approaches have named categories, but also have some of their own terminology as well. Due to scope limitations we will not go deeply into these approaches, but it is important to know what is out there to avoid confusion.

Relational Databases (sometimes called an RDBMS) store data in a way that uses relations among the data to link it in queries. As an example A person may have addresses and phone numbers and a relational database will typically store the three items (person, phone, and address) in separate data areas (called tables) but there will be index files setup that will allow a quick search from a person to the related phone numbers and address records.

The largest database type used is called "relational" and most DB developers have at least RDBMS knowledge.

Object Databases (sometimes called OODBMS) store data in a way that keeps related data together rather than using an index to find a record in another table. Using the person, phone, and address example from before.... The person data, phone numbers for the person, and addresses for the person will all be stored in a way that treats the data as a single object. This allows the data to be worked with as a group, rather than three separate retrieves (calls to the database engine).

Object Databases are not as common as RDBMS and tend to have harder to find and more expensive resources required.

Document databases store complex data such as a file or image together and work with the data as a whole document. Lotus Notes and Microsoft SitePoint are some examples of this.

Cloud and distributed databases are fairly similar to each other in that data is stored in a variety of locations, and using a variety of methods. This is usually done to allow for easy scaling to manipulate large amounts of data or the ability to store a wide variety of data types where each type is stored/retrieved in a manner best suited for that data type.

Cloud and distributed databases use a number of servers to spread out the work and allow for easy scalability.

NoSQL databases are a hot topic and are an attempt to make data retrieval easier than traditional SQL based engines. NoSQL (Not Only SQL) databases typically support SQL queries, but also allow for methods that make it easier to access data without a need to know how the data is stored. SQL typically requires one to know the table and column names and their relationships in order to retrieve data. NoSQL tries to abstract data retrieval methods and make it possible to use the database without learning SQL.

NoSQL databases allow for interaction with data without a need for SQL knowledge.

Data Warehouses are data storage approaches that are tuned for retrieval speed and calculations. These are also referred to as OLAP (Online analytical processing) rather than the other models, which are OLTP (Online Transaction Processing). A simplified definition of these acronyms is that OLAP databases are report based while OLTP databases are transaction (insert, update, delete) based. As an example, your amazon orders are likely entered into an OLTP, but daily/weekly/monthly reports most likely come from an OLAP database.

Data warehousing is used to provide quick ways to retrieve "report" data and groupings of the data.

Adoption

Databases of some sort are here to stay. Data is the heart of IT and storing, retrieving, and manipulating data is how the industry grows. Relational databases are still the king of the mountain. Oracle, Microsoft, and MySQL dominate the playing field. Postgres is a distant 4th place and would have to triple their market share to even approach MySQL. All of these are relational databases and not likely to lose market share anytime soon. Document database MongoDB is an up and comer, but still not likely to knock off relational databases. This is not as much about the approach as the standard types and amounts of data. Relational data covers all data, no matter how you look at it. A relationship can exist, but does not have to, so relational becomes almost a default way to store data.

Data and databases are used in all forms of IT projects with the most popular ones being relational DB systems.

Document and Object store databases are great for what they store, but there are a lot more small data items out there than documents and objects that group data. Document and Object databases can be thought of as a higher level data store than relational.

Small companies and start-ups are almost always going to use relational databases unless they are trying to make revolutionary data access part of their value proposition. You might see other database types, but they will be the ones with less development over head such as a document database or NoSQL.

Small companies and start-ups may use non-standard and niche databases as a form of competitive advantage.

NoSQL, and similar databases (Access, FileMaker, Excel as a form of database, etc), find ways to provide somewhat scalable data storage with a focus on ease-of- use/development. These show up in companies small and large. The usage of these tend to be smaller, proof of concept, or departmental projects whereas enterprise applications using these sort of databases are rare. These database types do tend to work good in tandem with a "higher end" database such as a large RDBMS for either reporting or accessing data in a disconnected state.

Data warehouses are often going to be found at companies with a large amount of data. This may be data from millions of sales, customers, or product combinations. The key is that data warehouses are almost always going to deal with "Big Data", where millions of records is still a small number.

Data warehouses are going to be found at companies with a large amount of data.

Market skill set

Database developers and administrators are often career focused and you can see a range of skill sets (and experience) in the market that shows it. There are entry level people working in databases that have no experience and no education ranging to those with doctorate degrees and no real world experience. As you move up the skill ladder it is not uncommon to see prospects with 5 to 10 years of experience and the veterans may have 30 or more years of database work on their resume.

Databases developers and administrators can have as much as 30 years experience or more at the most senior levels.

It is important to note that there is a big difference between database developers and database administrators. Database administrators will tend to be strong on working within the database in areas like tuning, stored procedures, etc. Database developers are stronger working with a database using external tools such as programming languages or writing queries.

Database developers and administrators are often treated as the same, but they have very different skill sets.

Certifications

Database certifications exist for every major database vendor and almost every lessor vendor. The value of the certifications vary a little from vendor to vendor, but they all tend to carry a lot of weight. Administrator certifications are more common and useful than developer certifications, but in both cases a certification can be a great way to bump a candidate up to a higher title/salary. Experience is important to database roles as you move into mid and senior levels, but certifications and educations can help bump an entry level resource up to a mid level position. This can be even more of a factor when major new releases of database systems are made available. In these cases the more experienced candidates have less advantage due to new features that are a large part of certification.

Certifications are available for nearly all levels and vendor types of databases.

Vendor sites are a great source for details about certifications, but there are also a number of third party companies that provide training, and various levels of certifications. The third party certifications tend to be vendor specific, although there are some that cover general (mostly SQL related) database skills. Another way to show expertise in database skills is a focused degree. There are a number of schools that offer general database degrees and even specific area of focus advanced degrees. These specialized degrees can replace an equal or higher number of years of real world experience in some cases. This means a master degree in Database Performance, for example, can easily be considered an equivalent of three to five years of experience (or more) at some companies.

The best place to start a search for certifications is on the vendor web site.

There are also third party certifications available to cover general topics like SQL.

Database certifications can be reached through a series of classes or tests. Even though the name may be the same, all certifications are not equal. When a certification is required, or listed, it is good to know what it takes to get that certification. Sometimes, how the certification was earned is almost as useful to know as that a certification was achieved. The journey taken can provide insight into how the certified prospect desires to use the certification and/or into what areas the prospect is most knowledgeable or comfortable. For example, the Microsoft MCSE: Data Platform certification requires five exams. Four of the exams are set, but one allows for replacement exams. The replacement exams can either be programming focused (C#), Server administrator focused, or service administrator focused. An MCSE with programming skills is going to be a different match than one with server administration skills. We will discuss this further in the section on complementary skills.

There are multiple paths to achieve database certification in some cases and the path taken is as important as the certification itself.

Suggested Database Certification Sites	
Oracle	http://education.oracle.com/pls/web_prod-plq-dad/ou_product_category.getPillarPage?p_pillar_id=2
Microsoft	https://www.microsoft.com/learning/en-us/certification-exams.aspx
MySQL	https://www.mysql.com/certification/

Check out the certificate page on the vendor web site or search for the vendor name and certification.

There are often a number of exams required for a database certification and these can be a key part of how valuable the certification is.

The number of exams that can be required for some of the Oracle and Microsoft certifications in particular can make it likely that multiple certifications are often earned as part of working through a reasonable series of certifications. This can even occur when targeting one specific certification. This can lead to certifications that are not really used. Always check with the candidate if they want to pursue a position their certification makes them a good fit for. Microsoft certifications seem to have the most cross over of exams and certifications, but other large (multi-test) certifications can suffer from the same lack of focus.

I personally have developer, database, and administrator certifications from Microsoft and have never had a desire to use the administrator certification. My work and skills lean heavily toward database and developer positions. I just got the administrator certification because it helped bump the company's number of certified staff. It was only one or two extra exams on top of the eight (I think) I had to take for the other certifications. This is just one case, but I have run into others in IT with similar stories of unused certifications.

Database Conversations

We have covered a lot of ground summarizing databases and touched a little bit on how to use this knowledge to find the best candidates for database positions, and now we can go a little deeper. In an area as wide ranging as databases we will have several clarifying questions to address. This is also an area where there are a lot of skills in one area that are a foundation for, or complementary to, others so we will spend some time on those as well.

Clarifying Questions

When dealing with database positions the first two things to clarify are: the system in use, and the role desired. The system question is a simple answer and is almost always included in any database related position. There are occasions when multiple database engines are listed and in these cases make sure to ask how the multiple systems are involved. There is typically a "main" engine used and then one or more "helper" systems although it may be a position that needs to support multiple applications and/or departments.

Always start by clarifying the system being used (vendor) and the desired role.

Sometimes more than one database is used and there will be a primary and secondary database engine.

As we have noted earlier, there are different ways to work with a database. Each of these may be called by many, overlapping names or titles, so you should always make sure responsibilities and details are clarified. This clarification alone can make a big impact on reducing the candidate pool to best fits for the position instead of wasting a lot of time vetting improper candidates.

Database Roles Summary	
Database Administrator	Installs, configures, and keeps a database system running. Typically maintains security and access issues as well. Focused on physical database design and aspects
Database Developer	Codes, tunes and populates the database. Focused on logical database design and aspects
Developer with DB skills	A "user" of the database that does some work within the engine itself, but usually makes calls from an application.

It is important to be able to distinguish between a DB administrator, a DB developer, and a developer with DB skills.

The role we call Database Administrator (DBA) is responsible for the physical database aspects such as installation, file management, configuration and other areas of the database are

where the role works. The DBA responsibility, in a nutshell, is to make the engine as stable and fast as possible, while making sure the database is available when needed. This is sort of like the goals of an auto mechanic in relation to a car.

A DBA is sort of like a mechanic for the database engine.

We use the phrase "Database Developer" when referring to positions responsible for developing/coding within the database engine. These roles perform tasks such as: writing stored procedures, tuning queries, designing the logical database structure and similar tasks. The roles are often referred to as a DBA, just as the one described above, but the skills and experience that makes the best fit for this role are very different.

A Database Developer is sort of like a driver of a car.

The third role is a "developer that works with a database." These roles may have the title of DBA, but typically are called "Database Developer" or just "Developer." These roles have responsibilities that center around building applications that utilize a database. The skills may include those listed in the "Database Developer" role just described but typically are limited to the ability to write queries and possibly includes some aggregation and computation.

A developer that works with databases is sort of like a passenger in a car.

Common Performance Tuning Strategies - Useful in any database engine -
Examine index usage
Look for unneeded joins
Look for unnecessary table scans
Look for data returned that is not needed
Look for translations/computations that are done too early.

Each of these roles are mixed together at times varying by company size and needs. The confusing part is that they often fall under a DBA or Database Developer title so it is always best to get some clarification as to what skills are really needed and the job expectations. This will help match skills and will also match desire as many candidates you will find that have a skill match across these three roles still have one or two areas they prefer to work in and are the best match for.

The roles sometimes overlap and can be very confusing to fill so the more details about daily tasks and expectations you can get from the job description, the better your chances of a good fit.

Crossover/Complementary Skills

The three roles described above have a lot of cross over and complementary skills. The ability to write queries is an entry level

requirement for any of the roles. The ability to write complex queries (inner/outer/left/right joins, inner selects, etc) also can apply to any of the roles and translates easily across them. Simple and complex query writing also tends to translate easily from database engine to engine. There is a little to be learned when going from relational, to object, to full text capable engines, but the concepts tend to be similar and easy to translate for mid level or more experienced candidates.

General SQL skills like writing queries (simple and complex) tend to translate across DB vendors.

Triggers and index creation has details to be learned with each engine, but this is another area where the skills tend to translate from engine to engine with a little work. In particular, a mid level or more experienced candidate with these skills will rarely take much time to ramp up to a new database engine.

Triggers and Index creation tends to be more vendor specific

Key Skills Summary	
Database Administrator	General Queries, Ability to read reports and log files to search for issues, physical DB design, network skills, Operating system skills, Index assessment
Database Developer	General Queries, Logical design, Stored procedures, Triggers, Index creation and assessment, Complex queries, Exposure to physical design
Developer with DB skills	Exposure to logical design, General Queries, exposure to complex queries, exposure to index creation, exposure to DB specific languages

Note that there is a lot of cross-over of skills across different DB roles.

Stored Procedure writing tends to be seen as the high water mark for database skills within a database (although tuning tends to also be seen this way). Design and creation of a database are higher skills, but can be considered external to database usage. Specific stored procedure skills vary from engine to engine as the language(s) used do vary, but the general concepts tend to be similar enough that a mid level or higher resource can translate their stored procedure skills to a new engine without too much ramp up time. The concepts will be the same just with a different syntax and a varying amount of bells and whistles. The differences are such that anyone with good Google skills or access to decent documentation will able to adapt and be immediately productive.

Stored Procedures are often considered the most complex of database work outside of design/creation.

Out of the administration skills, backup and installation skills are the least translatable. For these skills that are more server administration focused, the operating system becomes a factor so

Administration skills tend to be harder to translate to new vendors/systems.

it is more likely that a different engine on a familiar operating system will be easier to move to than one on a new operating system. In some cases simply moving the same database engine to a new operating system will require some learning for an administrator. We spend more time on these operating system issues in the Operating Systems Overview packet.

Administrators will cross over to a new engine easier than a new OS in some cases.

Performance tuning is very different from engine to engine and specific experience can command very high salaries or billing rates. There is some cross over of core theory behind database tuning that does tend to allow for a quick ramp up when dealing with a similar database style. Different styles of databases have different purposes and thus are often tuned differently. For example, relational database queries tend to perform best under a different set of rules than an object database. There is usually no trouble translating these skills at the senior, or higher, level but mid level and below resources might struggle.

Performance tuning is a very specific skill that can command a high salary or rate.

Weasels

Database related work tends to pay very well and thus attracts its share of weasels. In the database world a weasel tends to fall in one of two categories. There are the weasels that have been passed by as technology changed. These weasels used to be real good at their craft, and still may know some core information, but they tend to do things because “it has always been done that way” and will struggle in environments that need to take advantage of technological advances. The other weasels are the ones that have done a little database work and think they are better than they are. These weasels that lack self awareness may have the best of intentions or they may just be trying to use techno babble to get a better pay check.

High pay rates and compensation rates for database workers bring out the weasels.

The good news is that shining light on either of these weasel types just requires asking them to provide recent examples related to their database skills. The “past it” weasels will regale the listeners with stories about great feats from years ago and often even downplay the latest advances as “too much overhead” or “needlessly complicating things.” The “stretch” weasels might have some good stories with recent technology, but will show themselves to be bit players, to be indirectly involved, or might even come up empty, when asked for examples of more advanced work. I have seen this most often with great candidates that claim solid database skills and experience that have never written a

To verify a claim of skills ask for practical examples and prior experience. This will quickly weed out the weasels.

stored procedure or created a database from scratch. There are bits of knowledge you will only get if you have to perform these tasks and it is easy to underestimate that knowledge when you do not have it. Database skills are definitely an area where often "you don't know, what you don't know." Weasels will impress with terms, but experience is often the most important factor in database related roles.

Experience is the most important factor in database roles.

Screening Questions

Now that we have covered databases and related skills lets look at some example screening questions to use when trying to fill related positions.

- What database products have you used? This is a good starting question and may be answered with a glance at a resume. The greater the number of engines used, the better, even when only a single one is needed for a position

- What databases are you most comfortable with? Always a good follow up to the prior question, this can get a candidate talking about their experience and preferences to help you better match personality and chemistry

- Have you written stored procedures or functions? This is a more advanced question, but is a great one to weed out candidates when more advanced skills are needed.

- Have you installed [use the database type] before? This is aimed more at administrator roles, but still a good thing to know if administration skills are a plus

- Have you ever tuned [insert database name here]? If so what are some steps you had to take? This is to get at database tuning (of course) and it is a skill both a DBA and a database developer should have once they are mid level or higher. Examples they give are most often going to involve creation of indices, adjustments to table physical or logical design, and adjusting queries. You will probably hear terms like "table scan", "cartesian product", and maybe "clustered index." No need to know these in detail, just nod and smile.

- Explain a join and how it affects a query. This is a longer answer but should help you understand how a join grabs data from two tables in a query and joins always slow a query down.

Great Interview Questions

1. What DB have you used?
2. Which are you most comfortable with?
3. Have you written stored Procedures or functions?
4. Have you installed a specific database before?
5. Have you ever tuned a database or queries?
6. What is a join and how does it effect a query?
7. What is data normalization?

- What is data normalization and what are the normalization forms? Anyone with more than a few years of experience should be able to summarize data normalization (simplified answer: it reduces data duplication) and anyone that can give a good discussion of the forms off the top of their head knows databases very well, at least the theoretical side. There are usually 5 forms of normalization described and each one reduces the number of places data is stored to a greater degree. Storing state names in a lookup table and storing the id of the state rather than the full name is an example of a normalization step.

In general one can expect a database professional to start out being able to write queries and as they progress down the career path they should be able to write more complex queries, create indices and stored procedures, and improve general performance. By mid-levels of experience they will design tables and eventually full databases. You can tailor your questions with this in mind to quickly be able to order prospects from least to most experience without a need for much personal knowledge about databases.

Database professionals typically progress from writing simple queries to complex ones and into design and creation of databases.

Database Resources

Resources for more information

If you haven't had your fill, here are some more places to expand your knowledge:

http://www.transcender.com/?gclid=Cj0KEQjwiN6sBRDK2vOO_vaRs5cBEiQAfsnJCW99F6LDuOUCZijI1oQLVp412YHNtkYAABUBW8TN-w4aAons8P8HAQ&ef_id=VC6byQAAAQk3HJh0:20150704175552:s - A great third party source of certifications and training.

https://dev.mysql.com/downloads/ - Free download of the MySQL database and related documentation

https://go.oracle.com/LP=9865?elqCampaignId=9521&src1=ad:pas:go:dg:db&src2=wwmk14057296mpp011&SC=sckw=WWMK14057296MPP011&sckw=srch:oracle_database_download&mkwid=saKYvX9v3|pcrid|69738994459|pkw|%2Boracle%20%2Bdatabase%20%2Bdownload|pmt|b|pdv|c - Oracle 12c e-Book.

http://www.microsoft.com/en-us/server-cloud/products/sql-server-editions/sql-server-express.aspx - Microsoft SQL Server express download and information page.

Source Materials

http://dictionary.reference.com/browse/database - Dictionary definition of a database.

http://en.wikipedia.org/wiki/Database - Dictionary definition of a database.

https://www.microsoft.com/learning/en-us/certification-exams.aspx - Microsoft certification exams.

http://education.oracle.com/pls/web_prod-plq-dad/ou_product_category.getPillarPage?p_pillar_id=2 - Oracle Database certifications/training.

http://www.payscale.com/research/US/Job=Database_Developer/Salary - Database Developer salary information.

http://www.payscale.com/research/US/Job=Database_Analyst/Salary/885842bc/Experienced - Database Analyst Salary information.

http://www.payscale.com/research/US/Job=Back_End_Developer%2f_Engineer/Salary - Back End Developer information.

http://www.payscale.com/research/US/Job=Database_Administrator_(DBA)/Salary - Database Administrator information.