

SQL Server Interview Questions and Answers

For All Database Developers and
Developers Administrators

Pinal Dave

SQLAuthority.com

Vinod Kumar

ExtremeExperts.com

Table of Contents

TABLE OF CONTENTS	1
DETAILED TABLE OF CONTENTS	2
ABOUT THE AUTHORS	12
ACKNOWLEDGEMENT.....	14
PREFACE	14
SKILLS NEEDED FOR THIS BOOK	15
ABOUT THIS BOOK.....	16
DATABASE CONCEPTS WITH SQL SERVER	18
COMMON GENERIC QUESTIONS & ANSWERS.....	41
COMMON DEVELOPER QUESTIONS	57
COMMON TRICKY QUESTIONS.....	70
MISCELLANEOUS QUESTIONS ON SQL SERVER 2008	106
DBA SKILLS RELATED QUESTIONS	142
DATA WAREHOUSING INTERVIEW QUESTIONS & ANSWERS	142
GENERAL BEST PRACTICES	203

Detailed Table of Contents

TABLE OF CONTENTS	1
DETAILED TABLE OF CONTENTS	2
ABOUT THE AUTHORS	12
PINAL DAVE	12
VINOD KUMAR.....	13
ACKNOWLEDGEMENT.....	14
PREFACE	14
SKILLS NEEDED FOR THIS BOOK	15
ABOUT THIS BOOK.....	16
DATABASE CONCEPTS WITH SQL SERVER	18
WHAT IS RDBMS?.....	18
WHAT ARE THE PROPERTIES OF THE RELATIONAL TABLES?	18
WHAT IS NORMALIZATION?.....	19
WHAT IS DE-NORMALIZATION?	19
HOW IS THE ACID PROPERTY RELATED TO DATABASES?	20
WHAT ARE THE DIFFERENT NORMALIZATION FORMS?	20
WHAT IS A STORED PROCEDURE?	22
WHAT IS A TRIGGER?.....	23
WHAT ARE THE DIFFERENT TYPES OF TRIGGERS?	24
WHAT IS A VIEW?	25
WHAT IS AN INDEX?	26
WHAT IS A LINKED SERVER?.....	27
WHAT IS A CURSOR?	27
WHAT IS A SUBQUERY? EXPLAIN THE PROPERTIES OF A SUBQUERY?	28
WHAT ARE DIFFERENT TYPES OF JOINS?	29
EXPLAIN USER-DEFINED FUNCTIONS AND THEIR DIFFERENT VARIATIONS?.....	31
.....	32
WHAT IS THE DIFFERENCE BETWEEN A USER-DEFINED FUNCTION (UDF) AND A STORED PROCEDURE?	33

WHAT IS AN IDENTITY FIELD?	33
WHAT IS THE CORRECT ORDER OF THE LOGICAL QUERY PROCESSING PHASES?	34
WHAT IS A PRIMARY KEY?	35
WHAT IS A FOREIGN KEY?	35
WHAT IS A UNIQUE KEY CONSTRAINT?.....	36
WHAT IS A CHECK CONSTRAINT?	36
WHAT IS A NOT NULL CONSTRAINT?	36
WHAT IS A DEFAULT DEFINITION?.....	36
WHAT ARE CATALOG VIEWS?	37
POINTS TO PONDER FROM BEGINNING SQL JOES 2 PROS VOLUME 1 (ISBN: 1-4392-5317-X) (JOES2PROS.COM).....	37
COMMON GENERIC QUESTIONS & ANSWERS.....	41
WHAT IS OLTP (ONLINE TRANSACTION PROCESSING)?.....	41
WHAT ARE PESSIMISTIC AND OPTIMISTIC LOCKS?	41
WHAT ARE THE DIFFERENT TYPES OF LOCKS?	42
WHAT IS THE DIFFERENCE BETWEEN AN UPDATE LOCK AND EXCLUSIVE LOCK?	43
WHAT IS NEW IN LOCK ESCALATION IN SQL SERVER 2008?.....	43
WHAT IS THE NOLOCK HINT?.....	44
WHAT IS THE DIFFERENCE BETWEEN THE DELETE AND TRUNCATE COMMANDS?	44
WHAT IS CONNECTION POOLING AND WHY IS IT USED?	46
WHAT IS COLLATION?	46
WHAT ARE DIFFERENT TYPES OF COLLATION SENSITIVITY?.....	47
HOW DO YOU CHECK COLLATION AND COMPATIBILITY LEVEL FOR A DATABASE?	47
WHAT IS A DIRTY READ?	48
WHAT IS SNAPSHOT ISOLATION?.....	48
WHAT IS THE DIFFERENCE BETWEEN A HAVING CLAUSE AND A WHERE CLAUSE?	48
WHAT IS A B-TREE?	49
WHAT ARE THE DIFFERENT INDEX CONFIGURATIONS A TABLE CAN HAVE?.....	49
WHAT IS A FILTERED INDEX?	50
WHAT ARE INDEXED VIEWS INSIDE SQL SERVER?	50
WHAT ARE SOME OF THE RESTRICTIONS OF INDEXED VIEWS?.....	50
WHAT ARE DMVs AND DMFs USED FOR?.....	52
WHAT ARE STATISTICS INSIDE SQL SERVER?	53

POINTS TO PONDER FROM SQL QUERIES JOES 2 PROS VOLUME 2 ISBN: 1-4392-5318-8
(JOES2PROS.COM) 53

COMMON DEVELOPER QUESTIONS 57

WHAT IS BLOCKING? 57

WHAT IS A DEADLOCK? HOW CAN YOU IDENTIFY AND RESOLVE A DEADLOCK? 57

HOW IS A DEADLOCK DIFFERENT FROM A BLOCKING SITUATION? 58

WHAT IS THE MAXIMUM ROW SIZE FOR A TABLE? 58

WHAT ARE SPARSE COLUMNS? 59

WHAT ARE XML COLUMN-SETS WITH SPARSE COLUMNS? 59

WHAT IS THE MAXIMUM NUMBER OF COLUMNS A TABLE CAN HAVE? 60

WHAT ARE INCLUDED COLUMNS WITH SQL SERVER INDICES? 60

WHAT ARE INTERSECT OPERATORS? 61

WHAT IS THE EXCEPT OPERATOR USE FOR? 61

WHAT ARE GROUPING SETS? 61

WHAT ARE ROW CONSTRUCTORS INSIDE SQL SERVER? 62

WHAT IS THE NEW ERROR HANDLING MECHANISM STARTED IN SQL SERVER 2005? 63

WHAT IS THE OUTPUT CLAUSE INSIDE SQL SERVER? 64

WHAT ARE TABLE-VALUED PARAMETERS? 65

WHAT IS THE USE OF DATA-TIER APPLICATION (DACPAC)? 65

WHAT IS RAID? 66

WHAT ARE THE REQUIREMENTS OF SUB-QUERIES? 66

WHAT ARE THE DIFFERENT TYPES OF SUB-QUERIES? 67

WHAT IS PIVOT AND UNPIVOT? 67

CAN A STORED PROCEDURE CALL ITSELF OR ANOTHER RECURSIVE STORED PROCEDURE? HOW
MANY LEVELS OF STORED PROCEDURE NESTING ARE POSSIBLE? 67

POINTS TO PONDER FROM SQL ARCHITECTURE BASICS JOES 2 PROS VOLUME 3 ISBN:
1451579462 (JOES2PROS.COM) 68

COMMON TRICKY QUESTIONS 70

POINTS TO PONDER FROM SQL PROGRAMMING JOES 2 PROS VOLUME 4 ISBN:
1451579489 (JOES2PROS.COM) 103

*Pages 70 to 102 belong to
chapter **Common Tricky
Questions**. This chapter is
included in print book available
at
<http://bit.ly/sqlinterviewbook>*

MISCELLANEOUS QUESTIONS ON SQL SERVER 2008	106
WHAT ARE THE BASIC USES FOR MASTER, MSDB, MODEL, TEMPDB AND RESOURCE DATABASES?.....	106
WHAT IS THE MAXIMUM NUMBER OF INDICES PER TABLE?.....	107
EXPLAIN A FEW OF THE NEW FEATURES OF SQL SERVER 2008 MANAGEMENT STUDIO...	108
WHAT IS SERVICE BROKER?.....	110
WHAT DOES THE TOP OPERATOR DO?	111
WHAT IS A CTE?	111
WHAT DOES THE MERGE STATEMENT DO?	114
WHAT ARE THE NEW DATA TYPES INTRODUCED IN SQL SERVER 2008?	115
WHAT IS CLR?	118
DEFINE HIERARCHYID DATATYPES?	118
WHAT ARE TABLE TYPES AND TABLE-VALUED PARAMETERS?	119

WHAT ARE SYNONYMS?	119
WHAT IS LINQ?.....	120
WHAT ARE ISOLATION LEVELS?.....	120
HOW CAN YOU HANDLE ERRORS IN SQL SERVER 2008?.....	121
WHAT ARE SOME OF THE SALIENT BEHAVIORS OF THE TRY/CATCH BLOCK?.....	123
WHAT IS RAISEERROR?.....	124
WHAT IS THE XML DATATYPE?	125
WHAT IS XPATH?	125
WHAT IS TYPED XML?.....	126
HOW CAN YOU FIND TABLES WITHOUT INDEXES?	126
HOW DO YOU FIND THE INDEX SIZE OF A TABLE?	127
HOW DO YOU COPY DATA FROM ONE TABLE TO ANOTHER TABLE?	127
WHAT ARE SOME OF THE LIMITATIONS OF SELECT...INTO CLAUSE?	127
WHAT IS FILESTREAM IN SQL SERVER?	128
WHAT ARE SOME OF THE CAVEATS IN WORKING WITH THE FILESTREAM DATATYPE?	129
WHAT DO YOU MEAN BY TABLESAMPLE?.....	129
WHAT ARE RANKING FUNCTIONS?.....	130
WHAT IS ROW_NUMBER()?	131
WHAT IS A ROLLUP CLAUSE?.....	131
HOW CAN I TRACK THE CHANGES OR IDENTIFY THE LATEST INSERT-UPDATE-DELETE STATEMENTS FROM A TABLE?.....	131
.....	131
WHAT IS CHANGE DATA CAPTURE (CDC) IN SQL SERVER 2008?	132
WHAT IS CHANGE TRACKING INSIDE SQL SERVER?	132
HOW IS CHANGE TRACKING DIFFERENT FROM CHANGE DATA CAPTURE?	133
.....	133
WHAT IS AUDITING INSIDE SQL SERVER?.....	133
HOW IS AUDITING DIFFERENT FROM CHANGE DATA CAPTURE?.....	134
HOW DO YOU GET DATA FROM A DATABASE ON ANOTHER SERVER?	134
WHAT IS THE BOOKMARK LOOKUP AND RID LOOKUP?	134
WHAT IS THE DIFFERENCE BETWEEN GETDATE() AND SYSDATETIME() IN SQL SERVER 2008?	135
WHAT IS THE DIFFERENCE BETWEEN THE GETUTCDATE AND SYSUTCDATETIME FUNCTIONS?.....	135
HOW DO YOU CHECK IF AUTOMATIC STATISTIC UPDATE IS ENABLED FOR A DATABASE?	136

WHAT IS THE DIFFERENCE BETWEEN A SEEK PREDICATE AND A PREDICATE?	136
WHAT ARE VARIOUS LIMITATIONS OF VIEWS?	136
.....	137
WHAT ARE THE LIMITATIONS OF INDEXED VIEWS?	137
WHAT IS A COVERED INDEX?.....	138
WHEN I DELETE DATA FROM A TABLE, DOES SQL SERVER REDUCE THE SIZE OF THAT TABLE?	138
POINTS TO PONDER FROM SQL OF INTEROPERABILITY JOES 2 PROS VOLUME 5 ISBN: 1-4515-7950-0 (JOES2PROS.COM)	139
DBA SKILLS RELATED QUESTIONS	142

*Pages 142 to 173 belong to chapter **DBA Skills Related Question**. This chapter is included in print book available at <http://bit.ly/sqlinterviewbook>*

DATA WAREHOUSING INTERVIEW QUESTIONS & ANSWERS 142

*Pages 178 to 200 belong to chapter **Data Warehousing Q & A**. This chapter is included in print book available at <http://bit.ly/sqlinterviewbook>*

SQL WAIT STATS JOES 2 PROS: SQL PERFORMANCE TUNING TECHNIQUES USING WAIT STATISTICS, TYPES & QUEUES ISBN: 1-4662-3477-6 (JOES2PROS.COM)..... 201

GENERAL BEST PRACTICES **203**

ANNEXURE **207**

About the Authors

Pinal Dave

Pinal works as a Technology Evangelist with Microsoft India. He has been a part of the IT industry for more than seven years. During his career he has worked both in India and the US, primarily with SQL Server Technology - right from version 6.5 to its latest form. Pinal has worked on many performance tuning and optimization projects for high transactional systems. He received his Master of Science from the University of Southern California and a Bachelors of Engineering from Gujarat University.

Additionally, he holds many Microsoft certificates. He has been a regular speaker at many international events like TechEd, SQL PASS, MSDN, TechNet, and countless user groups. Pinal writes frequently on his blog <http://blog.SQLAuthority.com> on various subjects concerning SQL Server technology and Business Intelligence. His passion for the community drives him to share his training and knowledge. Before joining Microsoft, he was awarded the Microsoft Most Valuable Professional (MVP) in SQL Server Technology for three continuous years for his outstanding community service. He was also awarded the Community Impact Award – Individual Contributor for 2010. When he is not in front of a computer, he is usually travelling to explore hidden treasures in nature with his toddler daughter, Shaivi, and his very supportive wife, Nupur.

Pinal is also the co-author of two SQL Server books

- SQL Programming Joes 2 Pros: Programming & Development for Microsoft SQL Server 2008 (ISBN: 1451579489)
- SQL Wait Stats Joes 2 Pros: SQL server Performance Tuning Techniques Using Wait Types and Queues (ISBN: 1-4662-3477-6)

Vinod Kumar

Vinod Kumar has worked with SQL Server extensively since joining the industry over a decade ago. Working on various versions from SQL Server 7.0, Oracle 7.3 and other database technologies - he now works with Microsoft Technology Center (MTC) as a Technology Architect. With extensive database, BI and application background he currently helps customers maximize on the investments in technologies to solve real world business and integration problems. He has worked on various roles and projects involving development, migration, deployment, networking, architecture, testing, packaging, R&D for services and product based companies. He currently holds 26+ Microsoft Certifications on various technologies. Before joining Microsoft, he was a Microsoft MVP in SQL Server for more than 3 years.

In addition to his previous role inside Microsoft as Technology Evangelist (SQL Server), he is a known speaker at Technical conferences like Microsoft TechEd, MSDN, TechNet, GIDS. Vinod has more than 12+ years in computers and database systems since earning his Engineering degree from the College of Engineering, Guindy - Chennai. He has published numerous articles in SQL Server on multiple sites and currently writes most of his learning over his site and blog at:

<http://blogs.ExtremeExperts.com>. Feel free to follow on him on Twitter (@vinodk_sql).

Besides juggling all the activities, Vinod loves to watch movies, read books, explore food and enjoys spending time with his father (G.M.Sundaram) and supportive wife (Satya).

Acknowledgement

Rhonda Chesley is a SQL Server 2005 MCTS, an editor for Rick Morelan and all around geek wanna-be. She encourages girls of all ages to share her fascination of all things math and SQL.

Michael McLean has been working with Microsoft Technologies beginning in 1999 with Microsoft certifications in MCSE, MCSA, MCP+I, MCDBA and MCTS. Michael obtained his MCTS in Microsoft SQL 2008 Developer after attending Rick Morelan's school at MoreTechnology using the Joes2Pros books designed to help students pass Microsoft Exam 70-433 and now currently works at Microsoft in Redmond, WA on the Win7 Phone team.

Christopher Lennick earned a BS in Computer Science from Montana State University and has worked 26 years as an IT developer in various industries. He is currently employed by the largest natural and organic food distributor in the United States.

Preface

Today we are using computers for various activities, motor vehicles for traveling to places, and mobile phones for conversation. How many of us can claim the invention of micro-

processor, a basic wheel, or the telegraph? Similarly, this book was not written overnight. The journey of this book goes many years back with many individuals to be thanked for.

To begin with, we want to thank all those interviewers who reject interviewees by saying they need to know ‘the key things’ regardless of having high grades in class. The whole concept of interview questions and answers revolves around knowing those ‘key things’.

The core concept of this book will continue to evolve over time. I am sure many of you will come along with us on this journey and submit your suggestions to us to make this book a key reference for anybody who wants to start with SQL Server. Today we want to acknowledge the fact that you will help us keep this book alive forever with the latest updates. We want to thank everyone who participates in this journey with us.

Skills needed for this Book

We don’t expect our readers to be experts in the workings of SQL Server, nor is this book geared towards a complete novice. If you have had some experience working with databases (especially SQL Server) this book can be a wonderful refresher for the fundamentals. The Question & Answers format is a quick reference for the common questions asked during an interview. Feel free to use this as a cheat sheet prior to giving an interview and need a couple of questions to ask your candidate. If you are a student, please don’t read this book and think it is a “sure to pass” reference to crack an interview. Instead, this book can be used to

understand and get acquainted with some of the nuances and intricacies of SQL Server.

Having said all this, in God we trust and the rest we test. Feel free to test the concepts using actual code to reassert the concepts.

About this Book

As representatives of the IT community, we have certainly had our own experiences attending interviews where plenty of questions and doubts loomed...sometimes we failed miserably in the end! These stories have become pleasant (or not so pleasant) memories in our mind and this book will bring back some of those memories for sure. Once we analyzed the content of many, many interviews, we realized that most of them did not require a deep knowledge of the technical details of SQL, but they *did* require a solid understanding of the basics. Luckily, it is not necessary to know SQL inside and out to clear an interview. Subjects like “SQL Server” are so vast that it would take a lifetime to learn every detail of what it can do....even we learn something new every day!

There are a variety of careers for people who know SQL Server: Database Developer, Database Modelers, Database Architect, Database Administrator and many more. Hence, this book is geared towards demystifying and refreshing your memory of the fundamentals. Some of the concepts discussed are generic and are not tied to any specific version of SQL Server. That being said, most of the features are from SQL Server 2005 and 2008.

As we said before, this book is not a shortcut or a sure way to pass an interview. When faced with this big day, it is easy to get

overwhelmed and not know where to begin. This guide will help you prepare in an organized manner. This book can be that secret sauce you use to prepare.

Now this book will flow in a “Question & Answer” mode from start to finish to help you grasp concepts faster and get to the point quickly. Once you understand the concepts, it gets easier to see twists using that concept within a scenario and to ultimately solve them. Most companies have an interview strategy specific to scenarios relevant to their environment, needs and SLAs (Service Level Agreements).

Though each of these chapters are geared towards convenience we highly recommend reading each of the sections irrespective of the roles you might be doing since each of the sections have some interesting trivia about working with SQL Server. In the industry the role of accidental DBA’s (especially with SQL Server) is very common. Hence if you have performed the role of DBA for a short stint and want to brush-up your fundamentals then the upcoming sections will be a great review.

Education consists mainly of what we have unlearned. ~Mark Twain

Database Concepts with SQL SERVER

Life consists not in holding good cards but in playing those you hold well
- Josh Billings.

What is RDBMS?

Relational Data Base Management Systems (RDBMS) are database management systems that maintain data records and indices in tables. Relationships may be created and maintained across and among the data and tables. In a relational database, relationships between data items are expressed by means of tables. Interdependencies among these tables are expressed by data values rather than by pointers. This allows a high degree of data independence. An RDBMS has the capability to recombine the data items from different files, providing powerful tools for data usage. (Read more here <http://bit.ly/sqlinterview1>)

What are the properties of the relational tables?

Relational tables have the following five properties:

- Values are atomic.
- Column values are of the same kind.
- The sequence of columns is insignificant.
- The sequence of rows is insignificant.
- Each column must have a unique name.

What is normalization?

Database normalization is a data design and organizational process applied to data structures based on rules that help build relational databases. In relational database design, the process of organizing data to minimize redundancy is called normalization. Normalization usually involves dividing database data into different tables and defining relationships between the tables. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then retrieved through the rest of the database via the defined relationships.

The key traits for Normalization are eliminating redundant data and ensuring data dependencies.

What is de-normalization?

De-normalization is the process of attempting to optimize the performance of a database by adding redundant data. It is sometimes necessary because current DBMSs implement the relational model poorly. A true relational DBMS would allow for a fully normalized database at the logical level, while providing physical storage of data that is tuned for high performance. De-normalization is a technique to move from higher to lower normal forms of database modeling in order to speed up database access.

De-normalizing the database design allows for fewer joins with tables and foreign key requirements. This method is commonly used for Reporting and OLAP workloads.

How is the ACID property related to databases?

ACID (an acronym for Atomicity Consistency Isolation Durability) is a concept that database professionals generally look for while evaluating relational databases and application architectures. For a reliable database, all four of these attributes should be achieved:

Atomicity is an all-or-none rule for database modifications.

Consistency guarantees that a transaction never leaves your database in a half-finished state.

Isolation keeps transactions separated from each other until they are finished.

Durability guarantees that the database will keep track of pending changes in such a way that the server can recover from an abnormal termination and committed transactions will not be lost. (Read more here <http://bit.ly/sqlinterview3>)

What are the different normalization forms?

1NF: Eliminate repeating groups

Make a separate table for each set of related attributes and give each table a primary key. Each field contains at most one value from its attribute domain.

2NF: Eliminate redundant data

If an attribute depends on only part of a multi-valued key, then remove it to a separate table.

3NF: Eliminate columns not dependent on the key

If attributes do not contribute to a description of the key, then remove them to a separate table. All attributes must be directly dependent on the primary key. (Read more here <http://bit.ly/sqlinterview2>)

BCNF: Boyce-Codd Normal Form

If there are non-trivial dependencies between candidate key attributes, then separate them out into distinct tables.

4NF: Isolate independent multiple relationships

No table may contain two or more 1:n or n:m relationships that are not directly related.

5NF: Isolate semantically related multiple relationships

There may be practical constraints on information that justifies separating logically related many-to-many relationships.

ONF: Optimal Normal Form

A model limited to only simple (elemental) facts, as expressed in Object Role Model notation.

DKNF: Domain-Key Normal Form

A model free from all modification anomalies are said to be in DKNF.

Remember, these normalization guidelines are cumulative. For a database to be in the 3NF, it must first fulfill all the criteria of a 2NF and 1NF database.

What is a stored procedure?

A stored procedure (SP) is a named group of SQL statements that have been previously created and stored in the server database. Stored procedures are objects that do the work they are designed to do when you call upon them. You need to make sure they have what they need (the right values and parameters), so they can perform their important tasks. Stored procedures can act like views and select data, but they can also make updates, create objects, or even be set up to backup a database or perform other maintenance tasks.

Stored procedures accept input parameters so that a single procedure can be used over the network by several clients using different input data. When the procedure is modified, all clients automatically get the new version. Stored procedures reduce network traffic and improve performance. Stored procedures can be used to help ensure the integrity of the database logic. Typical system Stored Procedures are - sp_helpdb, sp_renamedb, sp_depends etc. (For a complete lesson on Stored Procedures see Chapters 6 and 7 of SQL Programming Joes 2 Pros Volume 4 ISBN: 1451579489)

Execution plan retention, reuse and improved security from SQL injection are some of the advantages of using SPs. (Read more here <http://bit.ly/sqlinterview4>)

What is a Trigger?

A trigger is a SQL procedure or SQLCLR Code that initiates an action when an event (like INSERT, DELETE or UPDATE) occurs on an object. Based on events which take place in your database, you can have SQL Server “listen” for just the ones that should signal when it’s time for actions to run automatically.

Triggers are stored in and managed by the DBMS. Triggers can be used to maintain the referential integrity of data by changing the data in a systematic fashion. A trigger cannot be called or executed directly; DBMS automatically fires the trigger as a result of a data modification to the associated table or in the case of DDL triggers to a DDL event in the database. Triggers are similar to stored procedures in that both consist of procedural logic that is stored at the database level. Stored procedures, however, are not event-driven and are not attached to a specific table as most triggers are. Stored procedures are explicitly executed by invoking a call to the procedure while triggers are implicitly executed by events. In addition, triggers can also execute stored procedures.

Nested trigger: A trigger can also contain INSERT, UPDATE and DELETE logic within itself; so when the trigger is fired because

of data modification, it can also cause data modification, thereby firing another trigger. A trigger is called a nested trigger when it is fired off from another trigger. (Read more here <http://bit.ly/sqlinterview5>)

SQL Server contains special triggers like login triggers and DDL triggers.

What are the different types of triggers?

There are three types of triggers. (For a complete lesson on Triggers see Chapters 2 and 3 of SQL Programming Joes 2 Pros Volume 4 ISBN: 1451579489)

1) DML trigger

There are two kinds of DML triggers

a. Instead of Trigger

Instead of Triggers are fired in place of the triggering action such as an insert, update, or delete.

b. After Trigger

After triggers execute following the triggering action, such as an insert, update, or delete.

2) DDL trigger

This type of trigger is fired against DDL statements like Drop Table, Create Table, or Alter Table. DDL Triggers are always after Triggers.

The data of DDL Triggers are captured in XML and called EventData which is available inside the trigger for logging at a later time.

3) Logon trigger

This type of trigger is fired against a LOGON event before a user session is established to the SQL Server.

What is a view?

A view can be thought of as a stored query accessible as a virtual table. It can be used for retrieving data as well as updating or deleting rows. Views in SQL Server provide a preset way to view data from one or more tables. They may also include aggregate fields (e.g., COUNT, SUM). Views allow your users to query a single object which behaves like a table and contains the needed joins and fields you have specified. In this way, a simple query (SELECT * FROM ViewName) can produce a more refined result which can serve as a report and answer business questions.

Rows updated or deleted in the view are updated or deleted in the table the view was created with. It should also be noted that as data in the original table changes, so does the data in the view-as views are the way to look at parts of the original table. The results of using a view are not permanently stored in the database. The data accessed through a view is actually constructed using a standard T-SQL select command and can come from one to many different base tables or even other views.

Two main purposes of creating a view are 1.) provide a security mechanism which restricts users to a certain subset of data and 2.) provide a mechanism for developers to customize how users can logically view the data.

What is an index?

An index is a physical structure containing pointers to the data. Indices are created in an existing table to locate rows more quickly and efficiently. It is possible to create an index on one or more columns of a table, and each index is given a name. The users can see the index name but cannot see the indices themselves; they are just used to speed up queries. Effective indices are one of the best ways to improve performance of a database application.

An Index can give you improved query performance because a seek action occurs for retrieving records from your table in a query. A seek means you were able to locate record(s) without having to examine every row to locate those record(s).

A table scan occurs when there is no index available or when a poorly created index exists on the table for a query running against that table. In a table scan, SQL Server examines every row in the table to satisfy the query results. Table scans are sometimes unavoidable, but on large tables, scans have a significant impact on performance (For a complete lesson on Indices see Chapters 8-11 of SQL Architecture Basics Joes 2 Pros Volume 3 ISBN: 1451579462).

Maximum number of Indices from SQL Server 2008 is 1 clustered + 999 non-clustered indices. (Read more <http://bit.ly/sqlinterview6>)

What is a linked server?

A linked server configuration enables SQL Server to execute commands against OLE DB data sources on remote servers. With a linked server, you can create very clean, easy-to-follow SQL statements that allow remote data to be retrieved, joined, and combined with local data. The ability to issue distributed queries and perform commands with transactions on heterogeneous sources is one of the benefits of using linked servers.

The system supplied stored procedures `sp_addlinkedserver` and `sp_addlinkedsrvlogin` are used to add new linked server(s). The stored procedure `sp_linkedservers` is used to list all the linked servers defined on the server. (Read more here <http://bit.ly/sqlinterview8>)

What is a cursor?

A cursor is a database object used by applications in the procedural logic to manipulate data in a row-by-row basis, instead of the typical SQL commands that operate on all or parts of rows as sets of data.

In order to work with a cursor, we need to perform these steps in the following order:

- Declare a cursor
- Open the cursor

- Fetch a row from the cursor
- Process the fetched row
- Close cursor
- Deallocate the cursor (Read more here <http://bit.ly/sqlinterview9>)

What is a subquery? Explain the properties of a subquery?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What are Different Types of Joins?

Inner join

A join that displays only the rows that have a match in both joined tables is known as an inner Join. This is the default type of join in the query and also in View Designer.

Generally the INNER JOIN option is the most common join in application queries.

Outer joins

A join that includes rows even if they do not have related rows in the joined table is an outer join. You can create three different variations of an outer join to specify the unmatched rows to be included:

- **Left outer join:** In a left outer join, all the rows in the first-named table, i.e. "left" table, which appears leftmost

in the JOIN clause, are included. An unmatched row in the right table appears as nulls in your result set.

- **Right outer join:** In a right outer join, all the rows in the second-named table, i.e. "right" table, which appears rightmost in the JOIN clause are included. An unmatched row in the left table will show nulls in your result set.

LEFT and RIGHT OUTER JOIN logic are opposite of each other. You can change either the order of the tables in the specific join statement or change the JOIN from left to right and get the same output.

- **Full outer join:** In a Full Outer Join, all of the rows all of the joined tables are included, whether they are matched or not.

Cross join

A cross join that does not have a WHERE clause produces the Cartesian product of the tables involved in the join. You can use a cross join to explore future possibilities. For example, at the beginning of a college semester, students may want to know what courses are required of them. Perhaps they have already satisfied some of those requirements. The requirements apply to all students regardless of the coursework they have done so far. A cross join simply returns all possible combinations of the record set data from the tables listed.

The size of a Cartesian product result set is *the number of rows in the first table multiplied by the number of rows in the second table*. One common example is when a company wants to combine each product with a pricing table to analyze each product at each price. (For a complete lesson on Cross Joins see Chapters 4 of Beginning SQL Joes 2 Pros Volume 1 ISBN: 1-4392-5317-X)

Self join

This is a special case when one table is joined to itself and aliasing the table name in one or two places to avoid confusion. A self join can be of any type (Inner Join or outer join), as long as the joined tables are the same. A self join is rather unique in that it involves a relationship with only one table. One common example is when a company has a hierarchal reporting structure wherein one member of the staff reports to another member of the staff. Self Joins are often used to show typical parts within other parts of the hierarchy. self joins can be an outer join or an inner join. (Read more here <http://bit.ly/sqlinterview11>)

Explain user-defined functions and their different variations?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What is the difference between a user-defined function (UDF) and a stored procedure?

UDFs can be used in SQL statements anywhere in the WHERE/HAVING/SELECT section, whereas stored procedures cannot. UDFs that return tables can be treated as another rowset. This can be used in JOINS with other tables. Inline UDFs can be thought of as views that take parameters and can be used in JOINS and other rowset operations.

Stored procedures can be used with INSERT Statements. (Read more <http://bit.ly/sqlinterview14>)

What is an identity field?

An identity (or AutoNumber) is a column that automatically generates numeric values. An identity field uniquely differentiates (or identifies) each record in a table. In the case of an identity field, that distinct value is an identifying number (e.g., InvoiceID, ProductID). The noteworthy feature of identity fields is that the identity property enforces data integrity by automatically generating the ID value each time you add a new record to the table.

There can be only one IDENTITY column in a given table inside SQL Server. A starting value and an increment value can be set, but most DBAs choose to start at 1. A GUID (Global Unique Identifier) column also generates numbers; the value of the identity cannot be controlled.

TRUNCATE TABLE resets the IDENTITY column to its base value. The DELETE command doesn't do this.

What is the correct order of the Logical Query Processing Phases?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What is a PRIMARY KEY?

A PRIMARY KEY constraint is a unique identifier for a row within a database table. A primary key prevents, duplicates, and ensures that all records have their own distinct values. Primary keys don't allow nulls, so you are guaranteed that each record has its own unique populated value.

Every table should have a primary key constraint to uniquely identify each row, and only one primary key constraint can be created for each table. The primary key constraints are used to enforce entity integrity. (For a complete lesson on all types of keys and constraints see Chapter 1 of SQL Programming Joes 2 Pros Volume 4 ISBN: 1451579489)

It is not possible to change the length of a column defined with a PRIMARY KEY constraint. If you need to change the length then you must first delete the existing PRIMARY KEY constraint and then re-create it with the new definition.

What is a FOREIGN KEY?

A FOREIGN KEY constraint prevents any actions that would destroy links between tables with the corresponding data keys. A simple way to think of a foreign key is that, essentially, it is another field which has a corresponding primary key field.

A foreign key in one table points to a primary key or unique key on another table. Foreign keys prevent actions that would change rows with foreign key values when there are no primary keys with that value. The foreign key constraints are used to enforce referential integrity.

What is a UNIQUE KEY Constraint?

A UNIQUE constraint enforces the uniqueness of the values in a set of columns; so no duplicate values are entered. The unique key constraints are used to enforce entity integrity as the primary key constraints.

Primary key is also a unique key internally, but cannot allow NULLs. unique keys on the other hand allow a single NULL but not multiple NULLs over the columns. (Read more here <http://bit.ly/sqlinterview16>)

What is a CHECK constraint?

A CHECK constraint is used to limit the values that can be placed in a column. CHECK constraints are most often used to enforce domain integrity. (Read more here <http://bit.ly/sqlinterview17>)

What is a NOT NULL constraint?

A not null constraint enforces that the column will not accept null values. Not null constraints are used to enforce domain integrity.

(Read more here <http://bit.ly/sqlinterview18>)

What is a DEFAULT definition?

A DEFAULT definition is used to add values into a column when values were omitted. The default value must be compatible with the data type of the column to which the DEFAULT definition applies. (For a complete lesson on Default Constraints see Chapter 1 of SQL Programming Joes 2 Pros Volume 4 ISBN: 1451579489)

DEFAULT values can be in integer and datetime fields but cannot be defined on timestamp and IDENTITY columns.

What are catalog views?

Catalog views return information that is used by the SQL Server database engine. Catalog views are the most general interface to the catalog metadata and provide the most efficient way to obtain, transform, and present customized forms of this information. All user-available catalog metadata is exposed through catalog views.

Points to Ponder from **Beginning SQL Joes 2 Pros Volume 1 (ISBN: 1-4392-5317-X) (Joes2Pros.com)**

1. A query is written in the SQL language and is a request for information from data in a database.
2. Microsoft SQL Server uses the Transact Structured Query Language (T-SQL).
3. The percent % symbol is the most common wildcard. This symbol represents any number of characters. For example, **WHERE Firstname like '%N'** would find a name that ends in N regardless of how long the name is. Examples may include Ann, MaryAnn and Dean among others.
4. The % sign can even represent zero characters. For example, **'%A%'** would find Alex and Lisa.
5. The SQL operator LIKE can be used to return a range of names, such as those beginning with a letter ranging from A to M. For example, **WHERE Firstname LIKE '[a-m]%'**

6. If you want to “exact match” a % symbol, like the name R%per!est and all other names with a percent symbol in them, surround the wildcard with square brackets. For example, **LastName LIKE '%[%]%'**
7. An inner join only returns a result set with perfectly matched values from fields in two or more tables.
8. An inner join is the default join type. If inner is omitted from the join clause of a query, SQL Server will assume it to be an inner join.
9. In a left outer join, the table named before the join might have records that appear even if SQL Server finds no matching records in the table listed after the LEFT OUTER JOIN clause.
10. The table listed after the RIGHT OUTER JOIN might have records that appear even if no matching records are found in the table on the left of the join.
11. When you alias a table, you use an abbreviation. SQL aliasing usually means using a shorter name than the original identifier.
12. A cross join creates or finds all possible entity combinations. The cross join does not need to use an ON clause.
13. Column names for a table must be unique. You can't have two fields named Hiredate in the same table.
14. A CREATE TABLE statement is a DDL statement. DDL means Data Definition Language. CREATE and DROP are DDL keywords.
15. Data Definition Language (DDL) statements handle the structure or design of database objects (e.g., databases

and tables) whereas Data Manipulation Language (DML) statements affect the actual data content. SELECT, INSERT, UPDATE and DELETE are four key DML keywords

16. Before SQL 2008 you could only insert 1 record with one insert statement. New to SQL 2008 is a feature called Row Constructors where you can insert many records at once with one insert statement.
17. SQLCMD is a command-line utility that allows you to run sql scripts or Ad-Hoc SQL queries.
18. BCP stands for Bulk Copy Program. BCP lets you perform data imports and exports using a command-line utility.
19. A transaction is a group of SQL statements treated as a single unit. Transactions ensure data integrity.
20. Transaction statements either all execute together or they don't at all.
21. If one statement can't run then the transaction is not committed.
22. A failed statement in a transaction means all data in the intermediate state gets discarded and none of the records will be committed.
23. The BEGIN TRANSACTION statement marks the beginning of a group of SQL statements in a transaction.
24. The COMMIT TRANSACTION marks the end of the transaction and saves all the changes to SQL's permanent storage.
25. If you want to read dirty data, you can use the READUNCOMMITTED table hint in your query.

26. The NOLOCK and READUNCOMMITTED table hints operate identically. Since NOLOCK is easier to type, it is used more often.
27. The advantage to the NOLOCK or READUNCOMMITTED table hints is your query runs without waiting for another process to release its locks.
28. To log into SQL Server, you need to create a server-level login. There are two types of server level logins (SQL Logins, Windows Logins).
29. Permissions can be manipulated with these DCL statements: GRANT, DENY, and REVOKE.
30. If you DENY a permission, it trumps any other permissions to that object or scope. For example, if you grant control and deny control, the user would have no access to the securable.
31. To remove an existing granted or denied permission, use the REVOKE keyword.

Common Generic Questions & Answers

Don't think outside the box. Think like there is no B O X. - unknown

What is OLTP (Online Transaction Processing)?

In OLTP (Online Transaction Processing systems), relational database design uses the discipline of data modeling and generally follows the Codd rules of data normalization in order to ensure absolute data integrity. Using these rules, complex information is broken down into its most simple structure (a table) where all of the individual atomic level elements relate to each other and satisfy the normalization rules.

What are pessimistic and optimistic locks?

Optimistic locking is a strategy where you read a record, take note of a version number and check that the version hasn't changed before you write the record back. If the record is changed (i.e. a different version to yours), then you abort the transaction and the user can re-start the transaction with the new data and update appropriately.

Pessimistic locking is when you lock the record for your exclusive use until you have finished with it. There are 4 levels of locking in the pessimistic isolation levels from lowest to highest. They are: read uncommitted, read committed, repeatable read, and serializable. At the serializable level (the

highest locking and isolation level) it has much better integrity than optimistic locking but requires you to be careful with your application design to avoid deadlocks.

What are the different types of locks?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What is the difference between an update lock and exclusive lock?

When exclusive lock is on any process, no other lock can be placed on that row or table. Every other process has to wait until the exclusive lock completes its tasks.

An update lock is a type of exclusive lock, except that it can be placed on the row which already has shared lock on it. Update lock reads the data of the row which has the shared lock. As soon as the update lock is ready to change the data it converts itself to the exclusive lock. (Read more here <http://bit.ly/sqlinterview19>)

What is new in lock escalation in SQL Server 2008?

Lock escalation is one of the lesser known phenomena inside SQL Server. Often times locking 1 row in a table cost less than locking the entire table. If you are updating thousands of records at once then SQL might find that is less costly to lock the table once rather than locking thousands of individual rows.

SQL Server uses this to minimize the overhead of locking too many structures by escalating the locks from just row locks to page locks to table locks. There is now a lock escalation option in the alter table of SQL Server 2008 which allows the disabling of lock escalation on that table.

What is the NOLOCK hint?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What is the difference between the DELETE and TRUNCATE commands?

The delete command removes the rows from a table on the basis of the condition that we provide a WHERE clause. Truncate will actually remove all of the rows from a table, and there will be no data in the table after we run the truncate command. (For a complete lesson on TRUNCATE and DELETE see Chapter 12 of SQL Queries Joes 2 Pros Volume 2 ISBN: 1-4392-5318-8)

TRUNCATE

- TRUNCATE is faster and uses fewer system and transaction log resources than DELETE.
- TRUNCATE removes the data by deallocating the data pages used to store the table's data, and only the page deallocations are recorded in the transaction log.
- TRUNCATE removes all the rows from a table, but the table structure, its columns, constraints, indexes, and permissions remain. You cannot use TRUNCATE TABLE on a table referenced by a FOREIGN KEY constraint. As TRUNCATE TABLE is not logged, it cannot activate a trigger.
- TRUNCATE cannot be rolled back unless it is used in a TRANSACTION. (To read more on this topic go here: <http://bit.ly/sqlinterview127>)
- TRUNCATE is a DDL Command.
- TRUNCATE resets the identity field of the table

DELETE

- DELETE removes one record at a time. If used with a predicate in a where clause and records an entry in the transaction log for each deleted row.
- If you want to retain the identity counter, use DELETE instead. If you want to remove table definition and its data, use the DROP TABLE statement.
- DELETE can be used with or without a WHERE clause
- DELETE activates triggers.

- DELETE can be rolled back.
- DELETE is a DML Command.
- DELETE does not reset the identity of the table.
(Read more here <http://bit.ly/sqlinterview21>)

What is connection pooling and why is it used?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

Only connections with the same configuration can be pooled. ADO.NET keeps several pools at the same time, one for each configuration.

What is collation?

Collation refers to a set of rules that determines how data is sorted and compared. Character data is sorted using rules that define the correct character sequence with options for specifying case sensitivity, accent marks, Kana character types,

and character width. (Read more here <http://bit.ly/sqlinterview23>)

After a collation has been assigned to any object or database, you cannot change the collation unless you drop and re-create the object/database.

What are different types of collation sensitivity?

Case sensitivity - A and a, B and b, etc.

Accent sensitivity - a and á, o and ó, etc.

Kana Sensitivity - When Japanese Kana, Hiragana, and Katakana characters are treated differently, it is called Kana sensitive.

Width sensitivity – When a single-byte character (half-width) and the same character represented as a double-byte character (full-width) are treated differently, it is width sensitive. (Read more here <http://bit.ly/sqlinterview24>)

How do you check collation and compatibility level for a database?

The following query can be used to see the collation and compatibility level your databases:

```
SELECT compatibility_level,  
collation_name  
FROM sys.databases  
WHERE name = 'YOUR DATABASE NAME HERE'
```

What is a dirty read?

A dirty read occurs when two operations, say, READ and WRITE occur together giving the incorrect or intermediate data. Suppose, User1 changed a row but did not commit the changes and User2 then reads the uncommitted data. The data may be wrong if User 2 does a rollback because the intermediate data may never have been committed. Because data in the intermediate state may never have been committed, it is considered a dirty read.

What is snapshot isolation?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

SQL Server extends the SQL-92 isolation levels with the introduction of the SNAPSHOT isolation.

What is the difference between a HAVING clause and a WHERE clause?

The HAVING clause specifies a search condition for a GROUP BY or an aggregate. The difference is that HAVING can be used only with the SELECT statement whereas the WHERE can be

used during update and delete operations. HAVING is typically used with a GROUP BY clause. The HAVING clause is used in an aggregate function or a GROUP BY clause in a query, whereas a WHERE Clause is applied to each row before they are part of the GROUP BY clause or aggregate function in a query. (Read more here <http://bit.ly/sqlinterview26>)

What is a B-tree?

The database server uses a B-tree structure to organize index information. B-tree generally has the following types of index pages or nodes:

- *Root node:* A root node contains node pointers to only one branch node.
- *Branch nodes:* A branch node contains pointers to two or more leaf nodes or other branch nodes.
- *Leaf nodes:* A leaf node contains index items and horizontal pointers to two or more leaf nodes.

What are the different index configurations a table can have?

A table can have one of the following index configurations.

- No indexes
- A clustered index only
- A clustered index with one non-clustered indexes
- A non-clustered index with no clustered index
- A clustered index with many non-clustered indices
- Many non-clustered indices with no clustered index

What is a filtered index?

A filtered index is used to index a portion of the rows in a table. This means it applies a filter on an INDEX which improves query performance, reduces index maintenance costs, and reduces index storage costs when compared with full-table indices. When we see an index created with a WHERE clause, then that is actually a Filtered Index. (Read more here <http://bit.ly/sqlinterview27>)

What are indexed views inside SQL Server?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What are some of the restrictions of indexed views?

There are plenty of restrictions for indexed views. If an Index is created on a view, then the definition of the view **must not** contain any of the following:

- ANY, NOT ANY
- OPENROWSET, OPENQUERY, OPENDATASOURCE
- arithmetic on imprecise (float, real) values
- OPENXML
- COMPUTE, COMPUTE BY
- ORDER BY (Read more here <http://bit.ly/sqlinterview28>)
- CONVERT producing an imprecise result
- OUTER join (Read more here <http://bit.ly/sqlinterview29>)
- COUNT(*) (Read more here <http://bit.ly/sqlinterview30>)
- reference to a base table with a disabled clustered index
- GROUP BY ALL
- reference to a table or function in a different database (Read more here <http://bit.ly/sqlinterview31>)
- Derived table (subquery in FROM list)
- reference to another view (Read more here <http://bit.ly/sqlinterview32>)
- DISTINCT
- ROWSET function
- EXISTS, NOT EXISTS
- self-join (Read more here <http://bit.ly/sqlinterview33>)
- expressions on aggregate results (e.g. SUM(x)+SUM(x))
- STDEV, STDEVP, VAR, VARP, AVG
- full-text predicates (CONTAINS, FREETEXT, CONTAINSTABLE, FREETEXTTABLE)
- A Subquery

- imprecise constants (e.g. 2.34e5)
- SUM on nullable expressions
- inline or table-valued functions
- table hints (e.g. NOLOCK)
- MIN, MAX
- text, ntext, image, filestream, or XML columns
- non-deterministic expressions
- TOP
- non-unicode collations
- UNION (Read more here <http://bit.ly/sqlinterview34>)
- Contradictions predicates making the view empty would be can be detected in SQL Server 2005 and newer (e.g. where 0=1 and ...)

Indexed views sometimes have very interesting behavior which can negate the whole purpose of the Indexed Views. (Read more here <http://bit.ly/sqlinterview35>)

What are DMVs and DMFs used for?

The DMVs (Dynamic Management Views) and DMFs (Dynamic Management Functions) were introduced in SQL Server 2005. It gives the database administrator information about the current state of the SQL Server machine on various aspects. From the basic definition, these dynamic management views and functions replace many of the DBCC command outputs and the pseudo table outputs. Hence, it is far easier to detect the health of SQL Server using these views and functions.

What are statistics inside SQL Server?

Statistics are the heart and soul of the SQL Server engine. SQL Server looks at the data in its tables long before you run your first SELECT statement. Because it's already done this, SQL Server knows how best to run a query when the time comes. Sampling of this data is stored in statistics, so the query optimizer can make the right decisions.

Without Statistics, the SQL Server Engine's Query Optimizer cannot decide the most optimal execution plan for the query. Statistics are used for SELECT, INSERT, UPDATE and DELETE operations. It is very important to keep statistics updated for SQL server to use the most efficient low resource execution plan. You can check the statistics on any table using the following command.

```
USE AdventureWorks
GO
sp_helpstats 'Person.Contact';
```

For a complete lesson on Statistics see Chapters 12 of SQL Architecture Basics Joes 2 Pros Volume 3 ISBN: 1451579462.

Points to Ponder from SQL Queries Joes 2 Pros Volume 2 ISBN: 1-4392-5318-8 (Joes2Pros.com)

1. The ORDER BY clause enables you to sort your query results. You can append the DESC (descending) and ASC (ascending) keywords to your ORDER BY clause.
2. If null values appear in your sort, they are first in ASC queries and last in DESC queries.

3. In SQL Server 7.0 and newer you can join up to 256 tables in a single query.
4. Databases often contain tables which exist for the sole purpose of allowing indirect relationships for Many-to-Many relationships between tables. These intermediary tables are known as Mapping Tables, Bridge Tables, or Junction Tables.
5. When filtering for nulls in queries, use the IS NULL or IS NOT NULL operators.
6. You can create expression fields in your query and base them upon other fields (like having a foreign currency price based upon the US price). An expression field is sometimes called a calculated field, a dynamic field, or a derived field.
7. The ANSI equivalent of GETDATE() is the property CURRENT_TIMESTAMP.
8. When using an aggregated function in your select list like Sum or Count you must have supporting aggregated language like GROUP BY or OVER.
9. The HAVING clause always appears after the GROUP BY clause. HAVING sets conditions on the aggregated values of the GROUP BY clause similar to the way WHERE interacts with SELECT.
10. The DISTINCT clause is useful to show all items in your query once, regardless of how many times they are listed. Use DISTINCT to eliminate duplicates or multiple listings of the same entity value when they are not relevant to your report. You can combine DISTINCT and HAVING to find aggregates that don't count repeating records multiple times.

11. A differing record count using a SELECT query vs. SELECT DISTINCT can be a rapid way to know if duplication exists.
12. With the OVER() clause you can integrate both actual base fields and aggregates in the same row. OVER() allows aggregation without requiring you to use a GROUP BY clause.
13. The OVER clause can aggregate rows across groups of another field using the PARTITION BY keyword. For example a grocery chain can find how sales of bananas compare to total produce sales instead of the entire food total sales. Leaving the OVER() blank causes it to apply the aggregation across all rows of the query.
14. The TOP results you see are based on your sorting order. For example, if you sort by SALES DESC you would see the largest sales at the top of the record set. If you choose SALES ASC, then the lowest sales would be listed at the top of the result set.
15. TOP is strictly a row limiter and does not evaluate or “read” data values. (It relies upon your sort order and slices off the specified number of records from the upper part of the result set.) When you add the keyword WITH TIES, it evaluates the data to include any tying values.
16. SQL Server has four ranking functions: RANK(), DENSE_RANK(), ROW_NUMBER(), and NTILE().
17. DENSE_RANK() counts ties and uses sequential number sequencing (no skipping, no gaps). Dense ranking closes the number gap caused by multiple rows having the same rank number. RANK() assigns tie records the same rank (three records tied for 6th place will all receive the rank of 6). After the tie is broken, RANK() assigns the next

record's row position as the rank (after the three records tied for 6th place, RANK() will assign 9 as the rank of the next record).

- 18.** The ROW_NUMBER() function is very similar to the RANK() and DENSE_RANK() functions. Use the ROW_NUMBER() function to number rows with no ties and no skipping.
- 19.** Like ROW_NUMBER, the NTILE() function ignores values, including ties. All records are handled in order and assigned to their respective groups. Records with the same value can be placed into separate groups. NTILE distributes rows into a specified number of groups.
- 20.** When using the UNION or UNION ALL operators you get record sets that are combined from multiple sources.
- 21.** UNION or UNION ALL operators require that all listed queries have the same number and type of fields in the same order.
- 22.** Since all the fields in the UNION have compatible (if not identical) data types, then it does not matter if the column names match. For example, you can union CustomerID and ProductID since both of these fields are the same data type (integer).
- 23.** In a UNION query only distinct records are selected. No duplicates.
- 24.** Using UNION ALL returns all records, including duplicates.
- 25.** Because UNION checks record by record to filter out duplicates, its processing time will be longer than a UNION ALL, which does not check for duplicates.

Common Developer Questions

The tragedy of life doesn't lie in not reaching your goal. The tragedy lies in having no goals to reach. - Benjamin Mays

What is blocking?

SQL Server blocking occurs when one connection places a lock on a table (or selected rows, pages, extent) and another connection attempts to read or modify the data when the lock is in effect. Another connection has to wait until the resources are released from the original connection which is holding the lock on the resources. Blocking often happens on the server when the system is under heavy transactional workload on a single resource. The way to resolve blocking is to identify the blocking statements which will then allow optimization of the blocking statements (re-write T-SQL, Indexing, or other configuration changes).

What is a deadlock? How can you identify and resolve a deadlock?

Deadlocking occurs when two user processes have locks on separate objects and each process is trying to acquire a lock on the object that the other process has locked. When a deadlock happens SQL server will then select the process with the least amount of overhead to rollback or abort. This way a deadlock is automatically resolved. There are multiple ways to

identify deadlocks i.e. Profile Deadlock Graph, DMV - sys.dm_tran_locks, and Extended Events.

How is a deadlock different from a blocking situation?

A deadlock occurs when two or more tasks permanently block each other by having a lock on a resource which the other task is trying to lock. In a deadlock situation, both transactions in the deadlock will wait forever unless the deadlock is broken. While in a standard blocking scenario, the blocked task will simply wait until the blocking task releases the conflicting lock.

Deadlocks can occur on resources other than database objects.

What is the maximum row size for a table?

The maximum bytes per row is 8086 (MSDN - <http://bit.ly/sqlinterview131>) in SQL Server 2008 R2. Additionally maximum bytes per varchar(max), varbinary(max), xml, text, or image column is 2GB ($2^{31}-1$). For a complete lesson on Date Types and Row Usage see Chapter 3 of SQL Architecture Basics Joes 2 Pros Volume 3 ISBN: 1451579462

SQL Server 2005 and later versions can handle more than 8086 bytes of data by moving the record to another page in the ROW_OVERFLOW_DATA allocation unit. In the original page it maintains a 24-byte pointer to this ROW_OVERFLOW_DATA allocation unit. (Read more here <http://bit.ly/sqlinterview36>).

What are sparse columns?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What are XML column-sets with SPARSE columns?

Tables that use sparse columns can designate a column to return all sparse columns in the table to XML data. A column set is like a calculated column in that the column set is not physically stored in the table. A column set differs from a calculated column in that the column set is directly updatable.

What is the maximum number of columns a table can have?

Maximum columns per table is 1024 in SQL Server 2008 R2.

A wide table is a table with a column set. This mean it contains SPARSE column(s), in this case the columns per table limit is 30,000. (MSDN - <http://bit.ly/sqlinterview132>)

What are INCLUDED columns with SQL Server indices?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What are INTERSECT operators?

The four multi-query operators are UNION, UNION ALL, INTERSECT, and EXCEPT. The INTERSECT operator introduced in SQL Server 2005 and later versions is used to retrieve the common records from both the left and the right query of the INTERSECT operator. The INTERSECT operator returns almost the same results as an INNER JOIN clause for all of the fields listed in the query. When using the INTERSECT operator the number and the order of the columns must be the same in all queries and the data type must be compatible. (Read more here <http://bit.ly/sqlinterview39>)

What is the EXCEPT Operator use for?

The EXCEPT operator is similar to the MINUS operation in Oracle. The EXCEPT query and MINUS query return all rows in the first query that are not found in the second query. For a complete lesson on Multiple Query Operators Chapters 8 of SQL Queries Joes 2 Pros Volume 2 ISBN: 1-4392-5318-8

Each SQL statement within the EXCEPT query (and MINUS query in Oracle) must have the same number of fields in the result sets with similar data types. (Read more here <http://bit.ly/sqlinterview40>)

What are GROUPING SETS?

The GROUPING SETS, ROLLUP, and CUBE operators are added to the GROUP BY clause. Though the results can be mimicked by using UNION ALL operators, these new constructs are far more efficient. There is a new function, GROUPING_ID(), that returns more grouping-level information than the existing

GROUPING() function. The non-ISO compliant WITH ROLLUP, WITH CUBE, and ALL syntax is being deprecated.

The new ROLLUP and CUBE syntax is only available in compatibility level 100.

What are row constructors inside SQL Server?

Transact-SQL is enhanced to allow multiple value inserts within a single INSERT statement. A simple construct is as follows –

```
INSERT INTO dbo.Persons (Name, Age)
VALUES ('Kumar', 35),
       ('Dave', 30)
```

You can do a double insert of data with one INSERT INTO statement using row constructors. Simply separate each group of values with a comma. The row constructor looks exactly like the double INSERT INTO except that you replace the INSERT with a comma. For a complete lesson on Row Constructors see Chapter 6 of Beginning SQL Joes 2 Pros Volume 1 ISBN: 1-4392-5317-X

The maximum number of rows that can be constructed using the table value constructor is 1000.

What is the new error handling mechanism started in SQL Server 2005?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What is the OUTPUT clause inside SQL Server?

The OUTPUT clause was introduced in SQL Server 2005, which is quite useful. The OUTPUT statement provides you with a confirmation copy of the records you just inserted, updated, deleted, or “upserted” using MERGE. The OUTPUT clause has access to inserted and deleted tables (virtual tables) just like triggers. The OUTPUT clause can be used to return values to the client clause. The OUTPUT clause can be used with INSERT, UPDATE, DELETE, or MERGE to identify the actual rows affected by these statements. For a complete lesson on the OUTPUT clause see Chapter 14 of SQL Queries Joes 2 Pros Volume 2 ISBN: 1-4392-5318-8

The OUTPUT clause can generate a table variable, a permanent table, or a temporary table. @@Identity will still work in SQL Server 2005, however you will find the OUTPUT clause very easy and powerful to use.” This gets the audience excited about being empowered instead of taking the authors

word for it. Here is an OUTPUT clause example (Read more here <http://bit.ly/sqlinterview42>).

What are table-valued parameters?

Table-valued parameters is a new feature introduced in SQL Server 2008. In earlier versions of SQL Server it was not possible to pass a table variable into a stored procedure as a parameter. In SQL SERVER 2008 we can use table-valued parameters to send multiple rows of data to a stored procedure or a function without creating a temporary table or passing in multiple parameters. Table-valued parameters are declared using user-defined table types. To use a table-valued Parameter we need to follow the steps shown below:

- Create a table type and define the table structure.
- Declare a stored procedure that has a parameter of table type.
- Declare a table type variable and reference the table type.
- Populate the variable using the INSERT statement.
- We can now pass the variable to the procedure. (Read more here <http://bit.ly/sqlinterview43>).

What is the use of data-tier application (DACPAC)?

The need for data-tier applications is to simplify development, deployment, and management of the database/data-tier objects that support multi-tier or client-server applications. DACPAC defines all of the database engine schema and instance objects, such as tables, views, and logins, required to support the application. The DAC operates as a single unit of

management through the development, deployment, and management lifecycle of the associated application. The DAC also contains policies that define the deployment prerequisites for the DAC.

A DAC can be deployed to instances of SQL Server 2008 R2 and SQL Azure.

What is RAID?

RAID (Redundant Array of Independent Disks) is a way of storing the same data in different places on multiple hard disks. By placing data on multiple disks, input/output operations can overlap in a balanced way, improving performance. The following are a few popular RAID type configurations used for database storage:

- RAID 0 – No Redundancy
- RAID 1 – Mirroring
- RAID 5 – Distributed Parity
- RAID 10 - Mirrored and Striped

What are the requirements of sub-queries?

- A sub-query must be enclosed in the parenthesis.
- A sub-query must be put on the right hand of the comparison operator.
- A sub-query cannot contain an ORDER BY clause.
- A query can contain more than one sub-query.

What are the different types of sub-queries?

- Single-row sub-query, where the sub-query returns only one row.
- Multiple-row sub-query, where the sub-query returns multiple rows.

What is PIVOT and UNPIVOT?

A pivot table can automatically sort, count, and total the data stored in one table or spreadsheet and create a second table displaying the summarized data. The PIVOT operator turns the values of a specified column into column names, effectively rotating a table. The UNPIVOT operator table is a reverse of the PIVOT operator. (To read more on this topic go here: <http://bit.ly/sqlinterview44>)

Can a stored procedure call itself or another recursive stored procedure? How many levels of stored procedure nesting are possible?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

Points to Ponder from SQL Architecture Basics Joes 2 Pros Volume 3 ISBN: 1451579462 (Joes2Pros.com)

1. A filegroup is a collection of datafiles which are managed as a single unit. SQL Server databases have a primary filegroup and may also have user defined secondary filegroups (like OrderHist).
2. You can only have one Primary filegroup per database but you can have as many user defined filegroups as you want.
3. Log files have a structure different from datafiles and cannot be placed into filegroups.
4. A schema is a namespace for database objects. In previous versions of SQL Server, database owners and schemas were conceptually the same object. Beginning in SQL Server 2005, owners and schemas are separate, and schemas serve as containers of objects.
5. A fully qualified name (FQN) is the complete object identifier. The FQN includes the server name, database name, schema name, and object name. The first three parts (server, database, and schema names) are known as the qualifiers of the object name, as they are used to differentiate the object from any other database object.
6. Database snapshots enable working with data as it appears at a point in time rather than reflecting the current status of the data. Snapshots are useful for reporting, development, and testing purposes.
7. The data portion of a row can contain the following elements:
 - Fixed length data
 - Null block

Common Developer Questions

- Variable block
 - Variable length data
- 8.** The data type defines the characteristic of the data that is stored in a column. In addition to the system-supplied data types, user-defined data types can be created for specific needs. User defined data types are also called Alias types.
- 9.** SQL Server 2008 has two specific spatial data types Geometry and Geography.
- 10.** The Geography data type stores round-earth latitude and longitude earth coordinates that represent points, lines, and polygons.
- 11.** The Geometry data type stores flat XY grid coordinates for points, lines, and polygons.
- 12.** A clustered index determines the physical organization of data in the table. Each table can have only one clustered index.
- 13.** When you create a primary key on a table, a clustered index is created by default unless you use the NONCLUSTERED argument.
- 14.** You can put a nonclustered index on a heap or on a clustered table. Nonclustered indexes are useful when users require multiple ways to search for data.
- 15.** You can implement nonclustered indices on heaps.
- 16.** Nonclustered indices are automatically rebuilt when:
- An existing clustered index on the table is dropped.
 - A clustered index on the table is created.
 - A column covered by the nonclustered index changes.

Common Tricky Questions

Success is not final, failure is not fatal: it is the courage to continue that counts - Winston Churchill

*Pages 70 to 102 belong to chapter **Common Tricky Questions**. This chapter is included in print book available at*

<http://bit.ly/sqlinterviewbook>

**Points to Ponder from SQL Programming Joes 2 Pros
Volume 4 ISBN: 1451579489 (Joes2Pros.com)**

1. Data integrity is the consistency and accuracy of the data which is stored in a database. A constraint performs data validation to help maintain database integrity by preventing invalid data from being entered.
2. You can create constraints using:
 - The CONSTRAINT keyword in the CREATE TABLE statement at the time you create the table.
 - The CONSTRAINT keyword in the ALTER TABLE statement after you have created the table.
3. Primary keys cannot accept null values, but unique indexes can accept 1 null value.
4. A check constraint restricts the values that users can enter into a particular column during INSERT and UPDATE statements.
5. You can temporarily disable a CHECK constraint. However, data integrity will not be enforced while the constraint is disabled.
6. Foreign keys are constraints that compare values between one column and another. Setting up a foreign key relationship enforces what is known as referential integrity.
7. If you don't want to check the existing data at the time you create the foreign key, then specify WITH NOCHECK.
8. A trigger is a special type of stored procedure that is not called directly by a user. A DML trigger is like a stored proc that executes when an INSERT, UPDATE, or DELETE event modifies data in a table.

9. There are two categories of triggers.
 - After Triggers – executed after the INSERT, UPDATE, or DELETE is performed. You can only define these on tables. AFTER triggers can be specified for tables but not for views.
 - Instead of Triggers – are executed in place of the usual triggering action. Unlike AFTER triggers, INSTEAD OF triggers can be specified for both tables and views.
10. Much like constraints, it is possible to enforce data integrity through triggers. However, you should use constraints whenever possible.
11. A TRUNCATE TABLE statement run against a table will unpopulate that table. However, no DML triggers will be fired off since TRUNCATE is a DDL statement.
12. What are the special memory resident tables available when dealing with triggers? Most people answer “Inserted, Updated, and Deleted.” However, there are only two tables: Inserted and Deleted.
13. A nested trigger is a trigger which executes a statement that causes an AFTER trigger to fire again.
14. A view is a virtual table whose contents are defined by a query. Views are database objects and are stored in your database, similar to tables, stored procedures, and functions.
15. The tables that make up a view are called “base tables.”
16. Views do not maintain separate copies of data. Therefore, when you are modifying records in a view, you are really modifying the records in the underlying base table.

17. A stored procedure is a named database object consisting of one or more lines of code. T-SQL statements run together in a single execution. A stored procedure is precompiled code that can be reused. You can also define your own custom stored procedures.
18. Table-valued parameters are new in SQL Server 2008. Table-valued parameters are a great way to pass in multiple rows of data at once, instead of just one value at a time.
19. A function is a SQL object stored in a database and consisting of T-SQL code that accepts parameters.
20. A user-defined function is a routine that you can create to accept parameters, perform a task, and return a result set.
21. Functions are similar to stored procedures in the way they work, but you must call a function using a SELECT statement or a WHERE clause within a SELECT statement.
22. If you want your procedure to raise an error defined by your own conditions, and this is not a system error, then you can call the RAISERROR() function. The first parameter can be a message number, text or a local @ variable. The message number corresponds to the message_id column in the sys.messages table.
23. If you use RAISERROR and specify text without a message number, you will get a message number of 50,000.
24. If you specify the message number and not the text, then you will get the text belonging to the message as listed in the Sys.Messages table.
25. You can't manually raise errors below 13000. Those may be raised only by the system.

Miscellaneous Questions on SQL SERVER 2008

If you can't make it good, at least make it look good.-Bill Gates

What are the basic uses for master, MSDB, model, tempdb and resource databases?

The **Master** database holds information for all the databases located on the SQL Server instance, and it is the glue that holds the engine together. Because SQL Server cannot start without a functioning master database, you must administer this database with care.

The **MSDB** database stores information regarding database backups, SQL Agent information, DTS packages, SQL Server jobs, PBM information and some replication information such as for log shipping.

The **tempdb** holds temporary objects such as global and local temporary tables and is a very important database for the instance. It is also the database that stores the version store when snapshot isolations are used. Each time SQL Server restarts the tempdb is copied from the model database.

Auto shrink is not allowed for **tempdb**. SQL Server 2008 R2 introduces **CHECKSUM** for tempdb.

The **model** database is essentially a template database used in the creation of any new user database created in the instance.

The **resource** database is a read-only database that contains all the system objects that are included in SQL Server. SQL Server system objects such as sys.objects are physically persisted in the resource database, but they logically appear in the sys schema of every database. The resource database does not contain user data or user metadata.

SQL Server cannot back up the resource database via maintenance plans or normal backup commands. Perform your own file-based backup by treating the mssqlsystemresource.mdf file as if it were any other binary file.

What is the maximum number of indices per table?

SQL versions up to SQL Server 2005:

1 clustered Index + 249 nonclustered Indices = 250 Indices.

For SQL Server 2008 and onwards:

1 clustered index + 999 nonclustered Indices = 1000 Indices.

(Read more here <http://bit.ly/sqlinterview71>)

Explain a few of the new features of SQL Server 2008 Management Studio.

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

Debugging capability was introduced with SQL Server 2008 Management Studio.

What is Service Broker?

Service Broker is a message-queuing technology in SQL Server that allows developers to integrate SQL Server fully into distributed applications. Service Broker is a feature which

provides functionality to SQL Server to be able to send asynchronous, transactional messages. It allows a database to send a message to another database without waiting for the response; so the application will continue to function if the remote database is temporarily unavailable. (Read more here <http://bit.ly/sqlinterview73>)

What does the TOP operator do?

TOP is a row-limiter and helps you limit the number of records affected by your DML statement. For a lesson on the TOP keyword see Chapter 6 of SQL Queries Joes 2 Pros Volume 2 ISBN: 1-4392-5318-8

The TOP operator is used to specify the number of rows to be returned by a query. The TOP operator has a new addition in SQL SERVER 2008 that allows it to accept variables as well as literal values. The TOP operator can be used with INSERT, UPDATE, and DELETE statements.

To use a variable / expression with the TOP operator it must be written inside parenthesis.

What is a CTE?

A CTE is the abbreviation for *Common Table Expression*.

*CTE Questions are on
Pages 112 to 114. These questions are
Included in print book available at
<http://bit.ly/sqlinterviewbook>*

What does the MERGE statement do?

The MERGE statement is new to SQL Server 2008 and is an excellent tool for handling table data. For years, database professionals have used the informal term *Upsert* with respect to bringing new data into an existing table. If the record is

brand new, you want your query logic to insert that entire record into the table. However, if the record already exists, you want your query to compare the old record with the new record and update only the incremental changes. That logical process has been formalized as the MERGE statement.

MERGE is a new feature that provides an efficient way to perform multiple DML operations in a single statement. In previous versions of SQL Server, we had to write separate statements to INSERT, UPDATE, or DELETE data based on certain conditions, but now, using a MERGE statement, we can include the logic of such data modifications in one statement that even checks when the data is matched, then just update the matched record, and when unmatched, insert it. For a lesson on using MERGE see Chapter 13 of SQL Queries Joes 2 Pros Volume 2 ISBN: 1-4392-5318-8.

One of the most important advantages of the MERGE statement is all the data is read and processed only once. (Read more here <http://bit.ly/sqlinterview75>)

While using the MERGE statement, take a look at the code written on DML TRIGGERS (if any), because these need to operate on a set of rows rather than a single row.

What are the new data types introduced in SQL SERVER 2008?

The GEOMETRY datatype: The GEOMETRY datatype is a system .NET common language runtime (CLR) datatype in SQL Server. This datatype represents data in a two-dimensional

Euclidean coordinate system. For a lesson on the GEOMETRY datatype see Chapter 5 of SQL Architecture Basics Joes 2 Pros Volume 3 ISBN: 1451579462.

The GEOGRAPHY datatype: The GEOGRAPHY and GEOMETRY datatype is a system .NET common language runtime (CLR) datatype in SQL Server. Many of the functions are the same as with GEOMETRY but GEOMETRY does have a few functions not found in the GEOGRAPHY datatype. Another difference between the two is that when you specify GEOGRAPHY, you are usually specifying points in terms of latitude and longitude. For a lesson on the GEOGRAPHY datatype see Chapter 5 of SQL Architecture Basics Joes 2 Pros Volume 3 ISBN: 1451579462.

SQL Server 2008 supports an out-of-box a set of methods for the **geometry** spatial data type. These include methods defined by the Open Geospatial Consortium (OGC) standard and a set of Microsoft extensions to that standard.

New DATE and DATETIME datatypes: SQL Server 2008 introduces four new data types related to date and time: DATE, TIME, DATETIMEOFFSET, and DATETIME2. For a lesson on date and time datatypes see Chapter 4 of SQL Architecture Basics Joes 2 Pros Volume 3 ISBN: 1451579462.

- **DATE:** The new DATE datatype just stores the date itself. It is based on the Gregorian calendar and handles years from 1 to 9999.

DATE datatype - Uses only 3-bytes (fixed) for storage of this data.

- **TIME:** The new TIME (*n*) datatype stores time with a range of 00:00:00.0000000 through 23:59:59.9999999. TIME supports seconds down to 100 nanoseconds. Precision setting changes are allowed with this datatype. The *n* in TIME(*n*) defines this level of fractional second precision from 0 to 7 digits of precision.

To support the new date and time data types SQL Server 2008 introduces new functions SYSDATETIME, SYSUTCDATETIME, SYSDATETIMEOFFSET, SWITCHOFFSET, and TODATETIMEOFFSET.

- **The DATETIMEOFFSET datatype:** DATETIMEOFFSET (*n*) is the time-zone-aware version of the datetime datatype. The name will appear less odd when you consider what it really is: a date + time + time-zone offset. The offset is based on how far behind or ahead you are from Coordinated Universal Time (UTC) time.
- **The DATETIME2 datatype:** It is an extension of the DATETIME datatype in earlier versions of SQL Server. This new datatype has a date range covering dates from January 1 of year 1 through December 31 of year 9999. DATETIME2 not only includes the larger date range, but also has a timestamp and the same

fractional precision that TIME datatype provides (down to 100 nanoseconds).

The standard DATETIME datatype could store data *January 1, 1753, through December 31, 9999* and had accuracy for time rounded to increments of .000, .003, or .007 seconds.

What is CLR?

The .NET framework can run managed code because it has the CLR. If SQL Server wanted to run managed code in its own environment, it would need its own .NET runtime. The SQL CLR is the engine SQL Server uses to run .NET code. SQL has its own built-in CLR. The SQL CLR runs managed code for SQL Server. In SQL Server 2005 and beyond, SQL Server objects such as user-defined functions can be created using such CLR languages. For a lesson on SQL CLR see Chapter 9 of SQL Interoperability Joes 2 Pros Volume 5 ISBN: 1-4515-7950-0.

This CLR language support extends not only to user-defined functions, but also to stored procedures and triggers. You can develop such CLR add-ons to SQL Server using Visual Studio. (Read more here <http://bit.ly/sqlinterview76>)

SQL Server 2008 lifts the limitation of CLR UDT's from 8000 bytes to 2GB.

Define HIERARCHYID datatypes?

The new HIERARCHYID datatype in SQL Server 2008 is a system-supplied CLR UDT that can be useful for storing and

manipulating hierarchies. It is internally stored as a VARBINARY that represents the position of the current node in the hierarchy. You can create indexes, query nodes, add siblings etc. just like any other data inside SQL Server.

What are table types and table-valued parameters?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

TVPs are read only in SQL Server 2008, and must be defined by using the READONLY keyword.

What are synonyms?

Also known as aliases, synonyms give you the ability to provide alternate names for database objects. For a lesson on Aliasing see Chapter 1 of Beginning SQL Joes 2 Pros Volume 1 ISBN: 1-4392-5317-X

You can alias object names; for example, you can alias the Employee table as Emp. You can also shorten names. This is especially useful when dealing with three and four part

names; for example, shortening server.database.owner.object to object. (Read more here <http://bit.ly/sqlinterview77>).

Simply alias the table name as another name in the FROM clause and then reuse the shorter name. Here is a SQL example:

```
SELECT *
FROM Location AS Loc INNER JOIN Employee
AS Emp
ON Loc.LocationID = Emp.LocationID
```

What is LINQ?

Language Integrated Query (LINQ) adds the ability to query objects using .NET languages. The LINQ to SQL object/relational mapping (O/RM) framework provides the following basic features:

- Tools to create classes (usually called *entities*) mapped to database tables.
- Compatibility with LINQ's standard query operations.
- The DataContext class with features such as entity record monitoring, automatic SQL statement generation, record concurrency detection, and much more.

What are isolation levels?

*This question is included in print book
available at
<http://bit.ly/sqlinterviewbook>*

SQL Server has an additional isolation called SNAPSHOT.

How can you handle errors in SQL SERVER 2008?

Ever since SQL 2005 SQL Server supports the use of TRY/CATCH constructs. This is for providing structured error handling. SQL Server introduced new and improved options for error handling beginning with SQL Server 2005. Prior versions did not include structured error handling. Structured error handling in SQL Server is similar to the way we approach

errors in real life. When something does not go exactly as we expected, we adapt and find other ways to accomplish our goals. For a lesson on TRY/CATCH blocks see Chapter 10 of SQL Programming Joes 2 Pros Volume 4 ISBN: 1451579489.

The job of a solution developer requires planning ahead and coding alternate pathways to keep our users and the application layer moving forward instead of stalling out when they encounter roadblocks. As analysts and application users, we have come to expect that application architects anticipate the majority of errors which our input could generate. Bugs which block the user from proceeding, or which force the user to exit and re-enter the application, are severe problems which should be caught and remedied during the test cycle.

TRY/CATCH blocks lets us build error handling at the level we need, in the way we need to by setting a region where if any error occurs, it will break out of the region and head to an error handler. The TRY block is where you place code which you think may raise an error. A TRY block is a code segment starting with a BEGIN TRY statement and ending with END TRY statement. If a statement sits inside a TRY block and raises an error, then the error gets passed to another part of SQL Server and not to the client. The TRY Block is aware that there is code which may fail.

The CATCH block serves as a contingency plan for failed code from the TRY block. In other words, if any statement raises a non-critical level 11 or higher severity error in the TRY block, it will not show the error from the calling code. It will run the

code you have set up in the CATCH block. The basic structure is as follows:

```
BEGIN TRY
<code>
END TRY
BEGIN CATCH
<code>
END CATCH
```

Errors that have a severity of 10 or lower / 20 or higher that cause the database engine to close the connection will not be handled by the TRY/CATCH block.

What are some of the salient behaviors of the TRY/CATCH block?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

Object name resolution errors and compile errors such as syntax errors that prevent a batch from executing are not caught in CATCH block.

What is RAISERROR?

SQL Server will raise errors when the code you have written cannot or should not execute. For example, a table should not be created if one with the same name already exists. Suppose you have a stored procedure named UpdateOneEmployee which changes one employee record at a time. The logic of this stored procedure will allow you to potentially update two employees with the same info. If it was against company policy to update more than one employee record at a time, it's extremely unlikely that anyone would ever attempt to update multiple records at once. However, because SQL Server has no restriction against updating one or many records in one transaction, you want to add a layer of protection to help enforce company policy. This is a case where you don't want SQL Server to allow this update, even though SQL Server doesn't define it as an error. To accomplish this goal, you can raise your own error message based on conditions which you define. For a lesson on RAISERROR blocks see Chapter 10 of SQL Programming Joes 2 Pros Volume 4 ISBN: 1451579489.

RAISERROR generates an error message and initiates error processing for the session. RAISERROR can either reference a user-defined message stored in the **sys.messages** catalog view or build a message dynamically. The message is returned as a server error message to the calling application or to an associated CATCH block of a TRY/CATCH construct. (Read more here <http://bit.ly/sqlinterview79>)

The old style syntax for RAISERROR (*Format: RAISERROR integer string*) syntax is deprecated.

What is the XML datatype?

Integers hold numbers with no decimal points, varchars hold strings of varying length, and the Geography datatype holds a position on the earth. Introduced in SQL Server 2005, the XML datatype holds and understands valid XML streams.

The **xml** datatype lets you store XML documents and XML fragments in a SQL Server database. An XML fragment is an XML instance that has a missing single top-level *root* element. You can create columns and variables of the **xml** datatype and store XML instances in them. The **xml** datatype and associated methods help integrate XML into the relational framework of SQL Server. For a lesson on the XML datatype see Chapter 5 SQL of Interoperability Joes 2 Pros Volume 5 ISBN: 1-4515-7950-0

What is XPath?

XPath uses a set of expressions to select nodes to be processed. The most common expression that you'll use is the

location path expression, which returns back a set of nodes called a *node set*. XPath can use both an unabbreviated and abbreviated syntax. The following is the unabbreviated syntax for a location path:

```
/axisName::nodeTest[predicate]/axisName::nodeTest[  
predicate]
```

For a lesson on XPath and XQuery see SQL of Interoperability
Joes 2 Pros Volume 5 ISBN: 1-4515-7950-0

What is typed XML?

We can create variables, parameters, and columns of the xml datatype - if we associate a collection of XML schemas with a variable, parameter, or column of xml datatype. In this case, the xml datatype instance is called typed XML. The fundamental advantage of using *typed XML* is we can perform some amount of validation constraint and even perform datatype validations.

How can you find tables without indexes?

Run the following query in the Query Editor.

```
USE <database_name>;  
GO  
SELECT SCHEMA_NAME(schema_id) AS schema_name  
      ,name AS table_name  
FROM sys.tables  
WHERE OBJECTPROPERTY(OBJECT_ID,'IsIndexed') = 0  
ORDER BY schema_name, table_name;  
GO
```

How do you find the index size of a table?

We can use the following query to find the size of the index.

```
EXEC sp_spaceused [HumanResources.Shift]
```

(Read more here <http://bit.ly/sqlinterview80>)

How do you copy data from one table to another table?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What are some of the limitations of SELECT...INTO clause?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What is Filestream in SQL Server?

Filestream allows you to store unstructured large objects (text documents, images, and videos) in the file system and have these files integrated within the database. Filestream basically integrates the SQL Server Database Engine with NTFS (New Technology File System); it basically stores the data in the varbinary(max) datatype. Using this datatype, the unstructured data is stored in the NTFS file system, and the SQL Server Database Engine manages the link between the Filestream column and the actual file located in the NTFS. Using Transact-SQL statements users can insert, update, delete and select the data stored in Filestream-enabled tables.

Filestream data is not encrypted even when transparent data encryption is enabled.

What are some of the caveats in working with the Filestream datatype?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

The 10-GB database size limit from SQL Server 2008R2 Express does not include the Filestream data container. It used to be 4GB in the previous versions of SQL Express.

What do you mean by TABLESAMPLE?

TABLESAMPLE allows you to extract a sampling of rows from a table in the FROM clause. The rows retrieved are random and they are not in any order. This sampling can be based on a percentage of number of rows. You can use TABLESAMPLE

when only a sampling of rows is necessary for the application instead of the full result set. (Read more here <http://bit.ly/sqlinterview82>)

What are ranking functions?

Ranking functions return a ranking value for each row in a partition. All the ranking functions are non-deterministic. For a lesson on Ranking Function see Chapter 7 of SQL Queries Joes 2 Pros Volume 2 ISBN: 1-4392-5318-8.

Each of the ranking functions also needs the OVER() clause. The different ranking functions are as follows:

***ROW_NUMBER () OVER ([<partition_by_clause>]
<order_by_clause>***

Returns the sequential number of a row within a partition of a result set, starting at 1 for the first row in each partition.

RANK () OVER ([<partition_by_clause>] <order_by_clause>

Returns the rank of each row within the partition of a result set. After a tie there will be numeric gaps in the next ranked number.

***DENSE_RANK () OVER ([<partition_by_clause>]
<order_by_clause>***

Returns the rank of rows within the partition of a result set, without any gaps in the ranking.

***NTILE (integer_expression) OVER ([<partition_by_clause>]
<order_by_clause>***

Distributes the rows in an ordered partition into a specified number of groups.

(Read more here <http://bit.ly/sqlinterview83>)

What is ROW_NUMBER()?

ROW_NUMBER() returns a column as an expression that contains the row's number within the result set. This is only a number used in the context of the result set; if the result changes, the ROW_NUMBER() will change.

What is a ROLLUP Clause?

The ROLLUP clause is used to do aggregate operations on multiple levels in hierarchy. If we want a sum on different levels without adding any new columns, then we can do it easily by using ROLLUP. We have to just add the WITH ROLLUP clause in the GROUP BY clause. (Read more here <http://bit.ly/sqlinterview84>)

How can I track the changes or identify the latest INSERT-UPDATE-DELETE statements from a table?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What is Change Data Capture (CDC) in SQL Server 2008?

Another innovative new feature in SQL Server 2008 is Change Data Capture (abbreviated as CDC). The concept is similar to Change Tracking but with a major difference. CDC tracks every field in your table(s) – not just the primary key fields. For a lesson on CDC see Chapter 12 of SQL Programming Joes 2 Pros Volume 4 ISBN: 1451579489.

Change Data Capture (CDC) records INSERTs, UPDATEs, and DELETEs applied to SQL Server tables and makes a record available of what changed, where, and when. The changed records in CDC are presented in simple relational *change tables* rather than in an esoteric chopped salad of XML. These changed tables contain columns that reflect the column structure of the source table you have chosen to track along with the metadata needed to understand the changes that have been made. (Read more here <http://bit.ly/sqlinterview85>)

What is change tracking inside SQL Server?

Change Tracking is a new and much anticipated feature in SQL Server 2008. Prior to SQL Server 2008, SQL DBAs had to code complex triggers and archive tables in order to track changes made to their database tables. For a lesson on Change Tracking see Chapter 11 of SQL Programming Joes 2 Pros Volume 4 ISBN: 1451579489.

Change tracking in SQL Server 2008 enables applications to obtain only changes that have been made to the user tables,

along with some information about those changes. With change tracking integrated into SQL Server, complicated custom change tracking solutions no longer have to be developed.

How is change tracking different from change data capture?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What is auditing inside SQL Server?

SQL Server audit offers features that help DBAs achieve their goals of meeting regulatory compliance requirements. SQL Server audit provides centralized storage of audit logs and integration with system center. SQL Server audit was designed with the following primary goals in mind:

- Security – The audit feature, and its objects, must be truly secure.
- Performance - Performance impact must be minimized.
- Management – The audit feature must be easy to manage.
- Discoverability - Audit-centric questions must be easy to answer.

How is auditing different from change data capture?

*This question is included in print book
available at
<http://bit.ly/sqlinterviewbook>*

How do you get data from a database on another server?

If you want to import data only through a T-SQL query, then use the OPENDATASOURCE function. To repeatedly get data from another server, create a linked server and then use the OPENQUERY function or use 4-part naming. If you are not using T-SQL, then it is better to use the import/export wizard, and you can save it as a SSIS package for future use. (Read more here <http://bit.ly/sqlinterview87>)

What is the bookmark lookup and RID lookup?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

Lookups are expensive, hence consider using the INCLUDE clause to make covering indexes.

What is the difference between GETDATE() and SYSDATETIME() in SQL Server 2008?

With the GETDATE() function, the precision is in milliseconds, and with the SYSDATETIME() function, the precision is down to 100 nanoseconds.

(Read more here <http://bit.ly/sqlinterview89>)

What is the difference between the GETUTCDATE and SYSUTCDATETIME functions?

These functions return data as UTC time (Coordinated Universal Time). In case of the GETUTCDATE(), the precision is in milliseconds. SYSUTCDATETIME() has a default precision of 7 digits after the seconds (aka nanoseconds). For a lesson on date and time datatypes see Chapter 4 of SQL Architecture Basics Joes 2 Pros Volume 3 ISBN: 1451579462.

`SYSDATETIME()`, `SYSUTCDATE()`, and `SYSDATETIMEOFFSET()` can be assigned to a variable in any one of the date and time datatypes.

How do you check if automatic statistic update is enabled for a database?

The following query can be used to determine if Automatic Statistic update is enabled:

```
SELECT is_auto_create_stats_on,  
is_auto_update_stats_on  
FROM sys.databases  
WHERE name = 'YOUR DATABASE NAME HERE'
```

(Read more here <http://bit.ly/sqlinterview90>)

What is the difference between a seek predicate and a predicate?

*This question is included in print book
available at*

<http://bit.ly/sqlinterviewbook>

What are various limitations of views?

*This question is included in print book
available at
<http://bit.ly/sqlinterviewbook>*

What are the limitations of indexed views?

For a lesson on Indexed Views see Chapter 5 of SQL Programming Joes 2 Pros Volume 4 ISBN: 1451579489.

Some of the limitations with Indexed views are:

- The UNION operation is not allowed in an indexed view. (Read more here <http://bit.ly/sqlinterview97>)
- We cannot create an Index on a nested view scenario which means we cannot create index on a view built from another view. (Read more here <http://bit.ly/sqlinterview98>)

- SELF JOINS are not allowed in indexed view. (Read more here <http://bit.ly/sqlinterview99>)
- OUTER JOINS are not allowed in an indexed view. (Read more here <http://bit.ly/sqlinterview100>)
- Cross database queries are not allowed in indexed views. (Read more here <http://bit.ly/sqlinterview101>)
- The view must be created using the WITH SCHEMABINDING option.
- ANSI_NULLS needs to be set for all existing tables that will be referenced in the view.
- Indexed views cannot contain text, ntext, image, filestream, or xml columns.

What is a covered index?

A covered index can satisfy a query just by its index keys without having the need to touch any data pages. When an Index covers the query in this way it is called covering index. It means that when a query is fired, SQL Server doesn't need to go to the table to retrieve the rows, but can produce the results directly from the index as the index covers all the columns used in query. (Read more here <http://bit.ly/sqlinterview102>)

When I delete data from a table, does SQL Server reduce the size of that table?

When data is deleted from any table, SQL Server does not reduce the size of the table right away; however, it marks those pages as free pages, showing that they belong to the table. When new data is inserted, they are put into those free pages first. Once those pages are filled up, SQL Server will

allocate new pages. If you wait for some time, the background process de-allocates the pages, finally reducing the page size. (Read more here <http://bit.ly/sqlinterview103>)

Points to Ponder from SQL of Interoperability Joes 2 Pros Volume 5 ISBN: 1-4515-7950-0 (Joes2Pros.com)

1. Metadata is data about your data. Metadata is information which helps to describe the properties and relationships of data.
2. Text files only contain data (no metadata). It's up to you to understand what the data means. XML is self-describing data, as it contains both data and metadata.
3. The FOR XML clause instructs SQL Server to return data as an XML stream rather than a rowset. The FOR XML clause is appended at the end of your SELECT statement.
4. The RAW option can be used with the ROOT or ELEMENTS keywords or both to customize your expected XML stream. The ROOT and ELEMENTS keywords are optional.
5. Raw and Auto modes both can use the ELEMENTS option. If ELEMENTS is not used, then both Raw and Auto will display your XML stream in attributes.
6. Path mode queries recognize a syntax called XPath to easily customize the layout.
7. SQL Server can turn table data into XML data. SQL Server can turn XML data into tabular data. XML shredding is the process of extracting data from XML streams and turning them into a tabular stream (e.g., a table).
8. Before you can process an XML document with T-SQL, you must parse the XML into a tree representation of the various nodes and store it within SQL Server's internal

cache using the `sp_XML_PrepareDocument` stored procedure.

9. The XML data type has a built-in method called `query()` which allows you to query for the parts of the XML you need.
10. XQuery was created primarily as a query language for getting data stored in an XML form.
11. XQuery is a query and functional programming language that is designed to query collections of XML data. The main purpose of XQuery is to get information out of XML databases.
12. XQuery is also capable of manipulating XML data by updating, inserting, and deleting data or elements. XPath is a subset of XQuery.
13. In SQL Server 2005 & 2008, the XML data type provides five methods:
 - `query()` – used to extract XML fragments from an XML data type.
 - `value()` – used to extract a single value from an XML document.
 - `exist()` – used to determine if a specified node exists. Returns 1 if it does exist, returns 0 if it doesn't exist.
 - `modify()` – updates XML data in an XML data type.
 - `nodes()` – shreds XML data into multiple rows.
14. Referencing relational columns in your XML field is known as “binding” the relational column.
15. An executable (.exe) file is known as an “out of process” assembly. Out of process assemblies contain enough information to launch or run on their own. An assembly

whose extension ends in .dll is known as an “in process” assembly.

16. SQL Server supports the use of in-process assemblies (DLL's) through the SQL CLR.
17. PowerShell goes beyond the cmd.exe shell that ships with windows. PowerShell is Microsoft's automation framework and command-line shell built on top of the .NET Framework.
18. SQL PowerShell is a shell that sends commands to SQL Server.
19. It's expected that, in the future, all Microsoft applications running on the Windows platform will be PowerShell aware.
20. PowerShell can run most of the old shell commands like dir, cd, and cls but also has its own commands that appear as a verb-noun combination.
21. PowerShell is made up of a command-line shell and its associated scripting language. Windows PowerShell talks to and is integrated with the Microsoft .NET Framework.
22. Windows PowerShell 2.0 was released with Windows 7 and Windows Server 2008 R2. This was released to manufacturing in August 2009.
23. Installing SQL in your system also installs new PowerShell Cmdlets.
24. The most used SQL PowerShell Cmdlet is Invoke-Sqlcmd.

DBA Skills Related Questions

If you want to test your memory, try to recall what you were worrying about one year ago today.- E. Joseph Cossman

*Pages 142 to 173 belong to chapter **DBA Skills Related Question**. This chapter is included in print book available at <http://bit.ly/sqlinterviewbook>*

Data Warehousing Interview Questions & Answers

*An expert is someone called in at the last minute to share the blame. -
Sam Ewing.*

*Pages 178 to 200 belong to
chapter **Data Warehousing Q &
A**. This chapter is included in print
book available at*

<http://bit.ly/sqlinterviewbook>

SQL Wait Stats Joes 2 Pros: SQL Performance Tuning Techniques Using Wait Statistics, Types & Queues ISBN: 1-4662-3477-6 (Joes2Pros.com)

1. Text files only contain data (no metadata). It's up to you to understand what the data means. XML is self-describing data, as it contains both data and metadata.
2. A wait stat happens when SQL Server wants to execute a task and it has to wait for resources to execute the task.
3. A task can be in one of three states:
 - Suspended
 - Runnable
 - Running
4. The key dynamic management view (DMV) that helps us to understand wait stats is `sys.dm_os_wait_stats`.
5. To reset the wait stats in `sys.dm_os_wait_stats` DMV following use the command:
 - `DBCC SQLPERF('sys.dm_os_wait_stats', CLEAR)`
6. If a session is waiting on a lock, then the `blocking_session_id` column of the `sys.dm_os_waiting_tasks` DMV will have the `session_id` of the lock.
7. The `sql_handle` of the `sys.dm_exec_requests` DMV points to the memory space of the actual SQL code being used by the task
8. You can get the SQL code from the `sys.dm_exec_sql_text` Dynamic Management Function (DMF) by passing in the SQL handle

9. A baseline is a measure of how things were going before you made any changes.
10. Parallel executions are important for computing since it allows more processing power (worker threads) at higher speed to be used for a single task.
11. There is an organizer-coordinator thread (thread 0), which creates organized worker threads into tasks which wait for all the worker threads to complete before the task can be closed.
12. When a big task has multiple threads starting at the same time, they won't always finish at the same time. When one or more of the threads will lag behind, thread 0 has to wait for all threads to finish, thereby creating the CXPACKET wait stat.
13. Note that not all the CXPACKET wait types are bad, since they only exist during parallel execution (i.e., which usually make things run faster).
14. Small tasks should not use multiple worker threads and will run more efficiently with just one thread.
15. Data-warehousing / Reporting servers tend to have very large tasks. They benefit from parallelism and will invariably have some CXPACKET wait types while large projects finish processing.
16. You will have little to no CXPACKET wait time with well-balanced parallelism or simple single CPU serial operation.
17. Mixed systems (OLTP + OLAP) present more of a challenge and the right balance must be found

General Best Practices

*Pages 203 to 206 belong to chapter **General Best Practices**. This chapter is included in print book available at <http://bit.ly/sqlinterviewbook>*

Please send errata, suggestions and feedbacks to
books@sqlauthority.com

Annexure

This book assumes at least an intermediate knowledge of SQL Server. This can be from a combination of experience and training. You need to already have a knowledge of queries, programming objects (like tables and stored procedures), and indexes. If you are new to the field but have done extensive studies, then this book assumes you have been exposed to the concepts in the first four of the five *SQL Joes 2 Pros* SQL development books:

Beginning SQL Joes 2 Pros: The SQL Hands-On Guide for Beginners
ISBN 1-4392-5317-X

SQL Queries Joes 2 Pros: SQL Query Techniques for Microsoft SQL Server 2008
ISBN 1-4392-5318-8

SQL Architecture Basics Joes 2 Pros: Core Architecture Concepts
ISBN: 1-4515-7946-2

SQL Programming Development Joes 2 Pros: Programming & Development for Microsoft SQL Server 2008
ISBN: 1-4515-7948-9

SQL Interoperability Joes 2 Pros: SQL 2008 Techniques with XML, C#, and PowerShell
ISBN: 1-4515-7950-0

SQL Wait Stats Joes 2 Pros: SQL Performance Tuning Techniques Using Wait Statistics, Types & Queues
ISBN: 1-4662-3477-6