# Part 1- Developer

## Querying Data with Transact-SQL – Skills (Exam 70-761)

### Manage data with Transact-SQL

### Create Transact-SQL SELECT queries

identify proper SELECT query structure, write specific queries to satisfy business requirements, construct results from multiple queries using set operators, distinguish between UNION and UNION ALL behavior, identify the query that would return expected results based on provided table structure, and/or data.

### Query multiple tables by using joins

write queries with join statements based on provided tables, data, and requirements; determine proper usage of INNER JOIN, LEFT/RIGHT/FULL OUTER JOIN, and CROSS JOIN; construct multiple JOIN operators using AND and OR; determine the correct results when presented with multi-table SELECT statements and source data; write queries with NULLs on joins

### Implement functions and aggregate data

construct queries using scalar-valued and table-valued functions; identify the impact of function usage to query performance and WHERE clause sargability; identify the differences between deterministic and non-deterministic functions; use built-in aggregate functions; use arithmetic functions, date-related functions, and system functions.

### Modify data

write INSERT, UPDATE, and DELETE statements; determine which statements can be used to load data to a table based on its structure and constraints; construct Data Manipulation Language (DML) statements using the OUTPUT statement; determine the results of Data Definition Language (DDL) statements onsupplied tables and dataQuery data with advanced Transact-SQL components (30–35%)

### Query data by using subqueries and APPLY

determine the results of queries using subqueries and table joins, evaluate performance differences between table joins and correlated subqueries based on provided data and query plans, distinguish between the use of CROSS APPLY and OUTER APPLY, write APPLY statements that return a given data set based on supplied data

### Query data by using table expressions

identify basic components of table expressions, define usage differences between table expressions and temporary tables, construct recursive table expressions to meet business requirements

### Group and pivot data by using queries

use windowing functions to group and rank the results of a query; distinguish between using windowing functions and GROUP BY; construct complex GROUP BY clauses using GROUPING SETS, and CUBE; construct PIVOT and UNPIVOT statements to return desired results based on supplied data; determine the impact of NULL values in PIVOT and UNPIVOT queries

### Query temporal data and non-relational data

query historic data by using temporal tables, query and output JSON data, query and output XML data

### Create database programmability objects by using Transact-SQL

create stored procedures, table-valued and scalar-valued user-defined functions, triggers, and views; implement input and output parameters in stored procedures; identify whether to use scalar-valued or table-valued functions; distinguish between deterministic and non-deterministic functions; create indexed views

### Implement error handling and transactions

determine results of Data Definition Language (DDL) statements based on transaction control statements, implement TRY…CATCH error handling with Transact-SQL, generate error messages with THROW and RAISERROR, implement transaction control in conjunction with error handling in stored procedures

### Implement data types and NULLs

evaluate results of data type conversions, determine proper data types for given data elements or table columns, identify locations of implicit data type conversions in queries, determine the correct results of joins and functions in the presence of NULL values, identify proper usage of ISNULL and COALESCE functions

## Developing SQL Databases (Exam 70-762)

### Design and implement a relational database schema

design tables and schemas based on business requirements, improve the design of tables by using normalization, write table create statements, determine the most efficient data types to use

### Design and implement indexes

design new indexes based on provided tables, queries, or plans; distinguish between indexed columns and included columns; implement clustered index columns by using best practices; recommend new indexes based on query plans

### Design and implement views

design a view structure to select data based on user or business requirements, identify the steps necessary to design an updateable view, implement partitioned views, implement indexed views

### Implement columnstore indexes

determine use cases that support the use of columnstore indexes, identify proper usage of clustered and non-clustered columnstore indexes, design standard non-clustered indexes in conjunction with clustered columnstore indexes, implement columnstore index maintenance

### Ensure data integrity with constraints

define table and foreign key constraints to enforce business rules, write Transact-SQL statements to add constraints to tables, identify results of Data Manipulation Language (DML) statements given existing tables and constraints, identify proper usage of PRIMARY KEY constraints

### Create stored procedures

design stored procedure components and structure based on business requirements, implement input and output parameters, implement table-valued parameters, implement return codes, streamline existing stored procedure logic, implement error handling and transaction control logic within stored procedures

### Create triggers and user-defined functions

design trigger logic based on business requirements; determine when to use Data Manipulation Language (DML) triggers, Data Definition Language (DDL) triggers, or logon triggers; recognize results based on execution of AFTER or INSTEAD OF triggers; design scalar-valued and table-valued user-defined functions based on business requirements; identify differences between deterministic and non-deterministic functions

### Implement transactions

identify DML statement results based on transaction behavior, recognize differences between and identify usage of explicit and implicit transactions, implement savepoints within transactions, determine the role of transactions in high-concurrency databases

### Manage isolation levels

identify differences between Read Uncommitted, Read Committed, Repeatable Read, Serializable, and Snapshot isolation levels; define results of concurrent queries based onisolation level; identify the resource and performance impact of given isolation levels

### Optimize concurrency and locking behavior

troubleshoot locking issues, identify lock escalation behaviors, capture and analyze deadlock graphs, identify ways to remediate deadlocks

### Implement memory-optimized tables and native stored procedures

define use cases for memory-optimized tables versus traditional disk-based tables, optimize performance of in-memory tables by changing durability settings, determine best case usage scenarios for natively compiled stored procedures, enable collection of execution statistics for natively compiled stored procedures Optimize database objects and SQL infrastructure (20–25%)

### Optimize statistics and indexes

determine the accuracy of statistics and the associated impact to query plans and performance, design statistics maintenance tasks, use dynamic management objects toreview current index usage and identify missing indexes, consolidate overlapping indexes

### Analyze and troubleshoot query plans

capture query plans using extended events and traces, identify poorly performing query plan operators, create efficient query plans using Query Store, compare estimated and actual query plans and related metadata, configure Azure SQL Database Performance Insight

### Manage performance for database instances

manage database workload in SQL Server; design and implement Elastic Scale for Azure SQL Database; select an appropriate service tier or edition; optimize database file and tempdb configuration; optimize memory configuration; monitor and diagnose scheduling and wait statistics using dynamic management objects; troubleshoot and analyze storage, IO, and cache issues; monitor Azure SQL Database query plans

### Monitor and trace SQL Server baseline performance metrics

monitor operating system and SQL Server performance metrics; compare baseline metrics to observed metrics while troubleshooting performance issues; identify differences between performance monitoring and logging tools, such as perfmon and dynamic management objects; monitor Azure SQL Database performance; determine best practice use cases for extended events; distinguish between Extended Events targets; compare the impact of Extended Events and SQL Trace; define differences between Extended Events Packages, Targets, Actions, and Sessions

# Part 2. Data Warehouse

## Implementing a Data Warehouse with Microsoft SQL Server (Exam 70-463)

### Design and implement dimensions

design shared/conformed dimensions; determine if you need support for slowly changing dimensions; determine attributes; design hierarchies; determine whether you need star or snowflake schema; determine the granularity of relationship with fact tables; determine the need for auditing or lineage; determine keys (business transactional or your own data warehouse/surrogate keys); implement dimensions; implement data lineage of a dimension table

### Design and implement fact tables

design a data warehouse that supports many to many relationships; appropriately index a fact table; using columnstore indexes; partitioning; additive measures; semi additive measures; non additive measures; implement fact tables; determining the loading method for the fact tables; implement data lineage of a fact table; design summary aggregation tables

### Define connection managers

plan the configuration of connection managers; package level or project level connection manager; define a connection string; parameterization of connection strings

## Design data flow

define data sources and destinations; distinguish blocking and non-blocking transformations; use different methods to pull out changed data from data sources; determine appropriate data flow components; determine the need for supporting Slowly Changing Dimensions (SCD); determine whether to use SQL Joins or SSIS lookup or merge join transformations; batch processing versus row by row processing; determine the appropriate transform to use for a specific task; determine the need and method for identity mapping and deduplicating; fuzzy lookup, fuzzy grouping and Data Quality Services (DQS) transformation; determine the need for custom data sources, destinations, and transforms; determine what to do with erroneous rows; determine auditing needs; trusted/authoritative data sources, including warehouse metadata; extracting only changed rows

## Implement data flow

debug data flow; use the appropriate data flow components; SQL / SSIS data transformation; create SSIS packages that support slowly changing dimensions; use the lookup task in SSIS; map identities using SSIS fuzzy lookup (advanced); specify a data source and destination; use data flows; different categories of transformations; read, transform and load data; understand which transforms to use to accomplish a specific business task; data correction transformation; performance tune an SSIS dataflow; optimize Integration Services packages for speed of execution; maintain data integrity, including good data flow

## Manage SSIS package execution

schedule package execution by using SQL Server Agent; execute packages by using DTEXEC; execute packages by using SQL Server Management Studio; implement package execution; plan and design package execution strategy; use PowerShell to execute script; monitor the execution using Management Studio; use DTEXECUI; ETL restartability

## Implement script tasks in SSIS

determine if it is appropriate to use a script task; extending the capability of a control flow; perform a custom action as needed (not on every row) during a control flow

## Design control flow

determine control flow; determine containers and tasks needed; determine precedence constraints; design an SSIS package strategy with rollback, staging and transaction control; decide between one package or multiple packages; determine event handlers; determine variables; determine parameters on package and project level; determine connection managers and whether they are package or project level; determine the need for custom tasks; determine how much information you need to log from a package; determine the need for checkpoints; determine security needs

## Implement package logic by using SSIS variables and parameters

user variables; variable scope, data type; implement parameterization of properties using variables; using variables in precedence constraints; referring to SSIS system variables; design dynamic SSIS packages; package configurations (file or SQL tables); expressions; package and project parameters; project level connection managers; variables; implement dynamic package behavior; configure packages in SSIS for different environments, package configurations (xmlconfiguration file, SQLServer table, registry entry; parent package variables, environment variable); parameters (package and project level); project connection managers; property expressions (use expressions for connection managers)

## Implement control flow

checkpoints; debug control flow; implement the appropriate control flow task to solve a problem; data profiling; use sequence containers and loop containers; manage transactions in SSIS packages; managing parallelism; using precedence constraint to control task execution sequence; creating package templates; using the execute package task

## Implement data load options

implement a full and incremental data load strategy; plan for an incremental update of the relational Data Mart; plan for loads into indexed tables; configure appropriate bulk load options; select an appropriate load technique (SSIS Destination versus T-SQL) and load partitioned tables

## Implement script components in SSIS

create an SSIS package that handles SCD Type 2 changes without using the SCD component; work with script component in SSIS; deciding when it is appropriate to use a script component versus a built in; source, transformation, destination component; use cases: web service source and destination, getting the error message Configure and deploy SSIS solutions (20–25%)

## Troubleshoot data integration issues

performance issues; connectivity issues; execution of a task or transformation failed; logic issues; demonstrate awareness of the new SSIS logging infrastructure; troubleshoot a failed package execution to determine the root cause of failure; troubleshoot SSIS package failure from an invalid datatype; implement break points; data viewers; profile data with different tools; batch cleanup

## Install and maintain SSIS components

software installation (IS, management tools); development box and server; install specifics for remote package execution; planning for installation (32- versus 64-bit); upgrade; provisioning the accounts; creating the catalog

## Implement auditing, logging, and event handling

audit package execution by using system variables; propagate events; use log providers; log an SSIS execution; create alerting and notification mechanisms; use Event Handlers in SSIS to track ETL events and errors; implement custom logging

## Deploy SSIS solutions

create and configure an SSIS catalog; deploy SSIS packages by using the deployment utility; deploy SSIS packages to SQL or file system locations; validate deployed packages; deploy packages on multiple servers; how to install custom components and tasks; deploy SSIS packages by using DTUTIL

## Configure SSIS security settings

SSIS catalog database roles; package protection levels; secure Integration Services packages that are deployed at the file system; secure Integration Services parameters, configuration

## Install and maintain data quality services

installation prerequisites; .msi package; adding users to the DQ roles; identity analysis, including data governance

### Implement master data management solutions

install Master Data Services (MDS); implement MDS; create models, entities, hierarchies, collections, attributes; define security roles; import/export; subscriptions

### Create a data quality project to clean data

profile Online Transaction Processing (OLTP) and other source systems; data quality knowledge base management; create data quality project; use data quality client; improve data quality; identity mapping and deduplicating; handle history and data quality; manage data quality/cleansing

# Part 3. Administrator

## Administering a SQL Database (Exam 70-764)

### Configure encryption

implement cell-level encryption, implement Always Encrypted, implement backup encryption, configure transparent data encryption, configure encryption for connections, troubleshoot encryption errors

### Configure data access and permissions

manage database object permissions, create and maintain users, create and maintain custom roles, configure user options for Azure SQL Database, configure row-level security, configure dynamic data masking

### Configure auditing

configure an audit on SQL Server, query the SQL Server audit log, manage a SQL Server audit, configure an Azure SQL Database audit, analyze audit logs and reports from Azure SQL Database

### Develop a backup strategy

back up very large databases, configure alerting for failed backups, back up databases to Azure, manage transaction log backups, configure database recovery models, configure backup automation

### Restore databases

perform piecemeal restores, perform page recovery, perform point-in-time recovery, restore file groups, develop a plan to automate and test restores

### Manage database integrity

implement database consistency checks, identify database corruption, recover from database corruption

### Monitor database activity

monitor current sessions, identify sessions that cause blocking activity, identify sessions that consume tempdb resources, configure the data collector

### Monitor queries

manage the Query Store, configure Extended Events and trace events, identify problematic execution plans, troubleshoot server health using Extended Events

### Manage indexes

identify and repair index fragmentation, identify and create missing indexes, identify and drop underutilized indexes, manage existing columnstore indexes

### Manage statistics

identify and correct outdated statistics, implement Auto Update Statistics, implement statistics for large tables

### Monitor SQL Server instances

create and manage operators, create and manage SQL Agent alerts, define custom alert actions, define failure actions, configure database mail, configure Policy-Based Management, identify available space on data volumes, identify the cause of performance degradation

### Implement log shipping

configure log shipping, monitor log shipping

### Implement AlwaysOn Availability Groups

configure Windows clustering, create an availability group, configure read-only routing, manage failover, create distributed availability groups

### Implement failover cluster instances

manage shared disks, configure cluster shared volumes

## Provisioning SQL Databases (Exam 70-765)

### Deploy a Microsoft Azure SQL database

Choose a service tier; create servers and databases; create a sysadmin account; configure elastic pools

### Plan for SQL Server installation

Plan for an IaaS or on-premises deployment; select the appropriate size for a virtual machine; plan storage pools based on performance requirements; evaluate best practices for installation; design a storage layout for a SQL Server virtual machine

### Deploy SQL Server instances

Deploy a SQL Server instance in IaaS and on-premises; manually install SQL Server on an Azure virtual machine; provision an Azure virtual machine to host a SQL Server instance; automate the deployment of SQL Server databases; deploy SQL Server by using templatesDeploy SQL Server databases to Azure virtual machinesMigrate an on-premises SQL Server database to an Azure virtual machine; generate benchmark data for performance needs; perform performance tuning on Azure IaaS; support availability sets in Azure

### Configure secure access to Microsoft Azure SQL databases

Configure firewall rules; configure Always Encrypted for Azure SQL Database; configure celllevel encryption; configure dynamic data masking; configure transparent data encryption (TDE)

## Configure SQL Server performance settings

Configure database performance settings; configure max server memory; configure the database scope; configure operators and alerts

## Manage SQL Server instances

Create databases; manage files and file groups; manage system database files; configure tempdb

## Manage SQL Storage

Manage SMB file shares; manage stretch databases; configure Azure storage; change service tiers; review wait statistics; manage storage pools; recover from failed storage

## Perform database maintenance

Monitor DMVs; maintain indexes; automate maintenance tasks; update statistics; verify database integrity; recover from database corruption