

**A QUICK LOOK AT**  
**T-SQL**  
**for**  
**MS-SQL Server**

**First Edition**

**Krishna Mohan Koyya**

**GLARIMY**

©2011 by Glarimy Technology Services. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the author.

Glarimy Technology Services  
106, Vars Casa Rosa, Pai Layout,  
BANGALORE – 560 016  
India

[www.glarimy.com](http://www.glarimy.com)

**A QUICK LOOK AT**  
**T-SQL**  
**for**  
**MS-SQL Server**

**First Edition**



# Contents

# About the Book

This book is part of the Glarimy's QuickLook series and by nature it is quick, short and to the point. These titles are meant for Java practitioners, learners and enthusiasts and not for sale.

## **On-line Version**

The whole content of this book is also available on-line in HTML format and PDF downloads at [www.glarimy.com/quicklook](http://www.glarimy.com/quicklook).

## **Code**

The code used as part of the illustrations in this book is available on-line as a jar file at [www.glarimy.com/quicklook](http://www.glarimy.com/quicklook). The jar files contain both source code and compiled classes.

## **Comments and Suggestions**

Your comments and suggestions are always appreciated to improve the quality and utility of these books. Please post your comments at [quicklook@glarimy.com](mailto:quicklook@glarimy.com).

# Preface

Welcome to *A Quick Look At T-SQL for MS-SQL Server*. This book is aimed to help the serious programmers who want to get into T-SQL, *quickly*. When I say serious programmers, I refer to the programmers who not only just read the technology but apply the theory and explore the technology in their own way.

The book designed and organized to understand the concepts *quickly* and to see the illustrations immediately and then to explore further beyond the concepts. Thus, each of the chapters is divided into three parts

- Understanding the Concepts
- Applying the Concepts
- Beyond the Concepts

True to their names, the part of *Understanding the Concepts* takes you through the concepts, quickly. It presents the problem that a feature of T-SQL addresses, the solution and other related facts. The part of *Applying the Concepts* deals with the code. It presents a full working code/queries supplemented with the explanation and output. This cements your understanding of the concepts. Once you are armed with the concepts, then the third part *Beyond the Concepts* takes you through few interesting facets and questions. Some of them are answered and some of them left for you to explore by writing code, going through the other documentation or just by thinking further.

All you need to use this book is SQL Express and OSQL command line utility. You are expected to be good at basic database and programming fundamentals. It would be easier for you to grasp the topics if you have good understanding of operating systems.

Hope this book helps in enhancing your interest and confidence in writing basic queries using T-SQL, again, *quickly*.

Krishna Mohan Koyya  
Bangalore  
21-July-2011

## About the Author

Krishna Mohan Koyya has been associated with the software industry for the last 14 years as a developer, teacher, trainer and author. His primary interests in technology include Java, object oriented architectures and Web 2.0.

As a software developer he worked with OpenView technology at Hewlett-Packard, Lucent's GSM OSS at Wipro Technologies and CiscoWorks NMS platform at Cisco Systems nearly for 10 years.

He taught Software Engineering at Sasi Institute of Engineering and Technology, Tadepalligudem for an year before taking up training and consulting as full time career in 2008.

As a trainer and consultant he handles most of the Java technologies ranging from JSE, JEE, Spring Framework, Struts Framework, JAX-WS, OSGi and etc and other technologies like Dojo framework, Mash-ups, UML and etc., Few of the clients to whom he delivered the services include Hewlett-Packard, Lucent-Alcatel, Intel, Sapient, IBM, Cap Gemini, Samsung, Oracle, Wipro, L&T and MindTree.

Currently he heads Glarimy Technology Services, Bangalore, India.

Academically he holds a bachelors in Electronics and Communications Engineering and masters in Computer Science and Technology, both from Andhra University, Visakhapatnam, India.

He can be reached at [krishna@glarimy.com](mailto:krishna@glarimy.com) or at [www.glarimy.com/krishna](http://www.glarimy.com/krishna)



## Database Management System

- Purpose
  - To store, maintain and retrieve data with ease
  - To maintain consistency and integrity of the data
  - To secure data
- Types
  - Flat-file based: Rudimentary, good for prototyping
  - Hierarchical: For naturally hierarchical 1-M data
  - Network: For naturally network M-M data
  - Relational: For data that confirms to relational algebra
  - Object: For naturally object oriented data
  - Object Relational: For mapping between relational and object oriented data

## Relational Database Management Systems

- Terminology
  - Field: column
  - Record: row or tuple
  - Table: relation
  - Cardinality: One-to-one, one-to-many, many-to-many
  - Type: Type of value at a cell
  - Domain: set of values that a cell can hold
  - Key: Means to identify a record
    - Candidate key: Any column with unique values
    - Primary key: Chosen candidate column for identification
    - Secondary key: Next chosen candidate column for identification
    - Composite key: Primary key with multiple columns
    - Foreign Key: Contains values from the primary key of other table
    - Surrogate Key: Primary key column with non-application data
- Normalization
  - Codd's rules for proper RDBMS
  - A process to maintain the data consistency and integrity (primary)
  - A process to save memory and process overheads (secondary)
    - All records must be uniquely identifiable
    - All non-key values must have dependency on the key
    - No non-key value must have dependency on another non-key
    - No grouping of elements in the same cell

- No duplication of more than one tuple
- Database Design
  - Using E-R diagrams
    - Entity as a rectangle
    - Ellipse as a field of the entity
    - Diamond as the relationship
    - Line with crow-foot as many-side of the relationship
    - Line as one-side of the relationship

### **Structured Query Language**

- A high-level language to interact with the RDBMS
- Standardized by ANSI and ISO
- Set of commands with prescribed syntax and semantics
- Primarily three classes of commands
  - Data Definition Language: To create, view, modify and delete the data structures (tables and etc)
  - Data Manipulation Language: To manipulate the data
  - Data Control Language: To manipulate access methods
- Operations on Databases
  - CRUD operations: Create, Read, Update and Delete

### **MS-SQL Server**

- The SQL implementation of Microsoft is Microsoft SQL Server
- In short, MS-SQL
  - Several popular versions like 2005, 2008
  - Several flavors like Express, Compact and etc.,
- Uses Transact SQL or T-SQL, an extension to ANSI SQL
- OSQL is the command line client to access the server
- Management Studio or Visual Studio are the GUI clients to access the server
- Supports batch scripting
  - Good for exporting
  - Slower than stored procedures
- MS-SQL Database Engine
  - MS-SQL Engine can maintain several databases
  - master is the default and system provided database

### **Transact-SQL or T-SQL**

- An extension to ANSI SQL. Some of the SQL syntaxes are not available
- Supports various data types
  - Exact Numeric types (bit, tinyint, smallint, int, bigint, decimal)
  - Approximate types (float, real)
  - Date and Time (date, time, datetime)
  - Character (char, varchar, text and nchar, nvarchar, ntext)
  - Binary (binary, image)
  - Others (cursor, table and etc)
- Supports various in-built functions
  - Aggregate Functions (min, max, sum, avg, count and etc.,)
  - Mathematical Functions (square, sqrt, abs, rand and etc., )
  - String Functions (upper, lower, trim, len and etc.,)
  - Date Functions (day, month, year, datediff, dateadd, datepar and etc.,)
  - Datatype Functions (identity, datalength)
  - System Functions (@@error, @@rowcount, host\_id, host\_name and etc., )
  - Meta-data Functions (object\_id, col\_length, object\_name and etc., )
  - Configuration Functions (@@language, @@servername and etc., )
  - Cursor Functions (@@fetch\_status, @@cursor\_rows and etc., )
- General Help commands
  - sp\_help: to view the table definition
  - sp\_helptrigger: to view the trigger definition
- DDL operations
  - To create, modify and drop database objects
    - Create | drop database
    - Create | alter | drop table
    - Create | alter | drop procedure
    - Create | alter | drop function
    - Create | alter | drop trigger
    - Create | alter | drop view

## Database Objects

- Tables
  - Defines the column properties and constraints
    - Properties: name, type, size and etc.,

- Constraints: primary key, unique, not null, default, check, foreign and etc.,
- Views
  - Table consisting results of SQL queries
  - May contain columns and rows from several tables
  - Read operations on views are allowed, but not write operations
- Stored Procedures
  - A named procedure using T-SQL
  - Gets compiled and stored for later invocation
  - May have any number of parameters either for input or for output
  - Gets executed with `exec <param>, <param>, ...` command
  - Can be invoked from other stored procedures as well
  - Procedural Constructs
    - Declaring parameters: using `@` notation
    - Declaring local variables: `DECLARE @<var_name> <type>`
    - Assigning values: `SET @<var> = <value> | (<select_query>)`
    - Decisions: `IF <condition> BEGIN <true_block> END [ELSE <false_block>]`
    - Loops: `WHILE <condition> BEGIN <body> END`
- Functions
  - A named procedure using T-SQL
  - Gets compiled and stored for later invocation
  - May have any number of parameters, all are inputs ONLY
  - May have only one return value
    - of any scalar type
    - of table type
- Cursors
  - Pointer to a fetched result set
  - Can be used as part of procedures and functions
  - Needs to be opened and closed
- Indexes
  - For faster access to the table data in search operations
  - Maximum number of variables as part of an index is 16
  - Maximum length of variables that are part of an index is 900 bytes
  - Two kinds:
    - Clustered Index

- Clustered index also sorts the table data
- Only one clustered index per table
- Primary Key is the natural index (clustered index)
- Non-clustered Index
  - Non-clustered index do not sort the data, it just holds pointers
  - Maximum of 249 non-clustered indexes on a table
  - Non-primary indexes (including unique) are non-clustered, by default
- Triggers
  - To be defined on a given table for insert or update or delete operations
  - Get executed after, by default, insert or update or delete operations
  - Contains any valid SQL
  - Create temporary inserted or updated or deleted tables, appropriately

### **Transactions**

- Database maintains a cache of data apart from the data on the hard-disk
- Operation of synchronizing the hard-disk data with the cache is called commit
- SQL Server commits the result of each query, immediately
- A transaction is to commit the results of several queries together
- Transaction operations
  - Begin Transaction: Commit would not be performed per query basis till the end of the transaction. Results of the queries are cached.
  - Commit Transaction: The hard-disk gets synchronized with the cache
  - Rollback Transaction: Results may be simply invalidated without commit, called rollback
- Transaction ACID Properties
  - Atomicity: All the queries of the transaction are treated unbreakable (all or none)
  - Consistency: In the beginning and at the end, the data must be consistent. May be inconsistent during the transaction.
  - Integrity: State of the records must not be accessible for outsiders as the data may undergo several changes during the transaction
  - Durability: Results of the transaction must be persisted.
- Transaction Lock Modes
  - Shared Lock: Used for read operations
  - Exclusive Lock: Avoids multiple updates on the same resource at the same time

### **T-SQL Syntax**

1. Connecting to the database server from command prompt  
`osql [-E|-U <user_name>] [-P <password>] [-d <database>]`
2. Listing out all the databases  
`select name from sysdatabases  
go`
3. Creating your own database  
`create database <database_name>`  
  
`create database glarimy  
go`
4. Using a database  
`use <database_name>`  
  
`use glarimy  
go`
5. Listing all the tables in the database  
`select name from sysobjects where type='U'  
go`
6. Creating your own table with basic features  
`create table <table_name> (  
 <column_name> <column_type>,  
 . . .  
)`  
  
`create table books (  
 isbn int,  
 title varchar(32),  
 price decimal(5,2),  
 entry_date datetime,  
 type varchar(32)  
)  
go`
7. Inserting records in to the table  
`insert into <table_name> values (<value>, <value>, <value>, ...)`  
  
`insert into books values (12345, 'T-SQL', 45.75, '7/22/2011',  
'Reference')  
go`
8. Inserting values for only few columns of a table  
`insert into <table_name> (<column_name>, ...) values (<value>, ...)`

```
insert into books (isbn, title, entry_date) values (2345, 'History',
'10/23/2007')
go
```

9. **Dropping a table**  
**drop table <table\_name>**

```
drop table books
go
```

10. **Creating table with primary key**

```
create table <table_name> (
  <column_name> <column_type> [<primary key>],
  . . .
)

create table books (
  isbn int primary key,
  title varchar(32),
  price decimal(5,2),
  entry_date datetime,
  type varchar(32)
)
go
```

11. **Creating table with basic constraints**

```
create table <table_name> (
  <column_name> <column_type>
  [primary key] [not null] [unique] [default <value>],
  . . .
)

create table books (
  isbn int primary key,
  title varchar(32) not null unique,
  price decimal(5,2) not null default 0,
  entry_date datetime not null,
  type varchar(32) not null default 'Reference'
)

```

12. **Creating table with check constraints**

```
create table <table_name> (
  <column_name> <column_type>
  [primary key] [not null] [unique] [default <value>]
  [check(<constraint>)] [check(<constraint>)] ...,
  . . .
  [check(<constraint>)],
  . . .
)
```

```

)

create table books (
    isbn int primary key,
    title varchar(32) not null unique,
    price decimal(5,2) not null default 0 check(price >= 0),
    entry_date datetime not null,
    type varchar(32) not null default 'Reference'
    check (type='Reference' or type='Text' or type='Journal')
)

```

### 13. Creating table with foreign key constraint

```

create table <table_name> (
    <column_name> <column_type>
        [primary key] [not null] [unique] [default <value>]
        [check(<constraint>)] ...
        [references <foreign_table_name(<foreign_key_column_name>)>]
        . . .
        . . .
        [check(<constraint>)],
        . . .
)

create table authors (
    name varchar(32) primary key,
    phone_number bigint not null,
    email_id varchar(32)
)

create table books (
    isbn int primary key,
    title varchar(32) not null unique,
    author_name varchar(32) references authors(name),
    price decimal(5,2) not null default 0 check(price >= 0),
    entry_date datetime not null,
    type varchar(32) not null default 'Reference'
    check (type='Reference' or type='Text' or type='Journal')
)

```

### 14. Creating table with auto incrementing facility

```

create table <table_name> (
    <column_name> <column_type>
        [primary key] [identity(<base, increment>)],
        [not null] [unique] [default <value>]
        [check(<constraint>)] ...
        [references <foreign_table_name(<foreign_key_column_name>)>]
        . . .
        . . .
        [check(<constraint>)],
        . . .
)

```



```

create table authors (
    name varchar(32) primary key,
    phone_number bigint not null,
    email_id varchar(32)
)

create table books (
    isbn int primary key identity(1,1),
    title varchar(32) not null unique,
    author_name varchar(32) references authors(name),
    price decimal(5,2) not null default 0 check(price >= 0),
    entry_date datetime not null,
    type varchar(32) not null default 'Reference'
    check (type='Reference' or type='Text' or type='Journal')
}

```

#### 15. Adding a column

```

alter table <table>
add <column_name> <column_type> [not null]

alter table <books>
add pages int

```

#### 16. Dropping a column

```

alter table <table>
drop column <column_name>

alter table books
drop column pages

```

#### 17. Modifying a column size or nullability

```

alter table <table_name>
alter column <column_name> <column_type> [null | not null]

alter table books
alter column pages int not null

```

#### 18. Adding a primary key

```

alter table <table_name>
add primary key <column_name>

alter table books
add primary key(title)

```

#### 19. Adding a unique constraint

```

alter table <table_name>
add unique(<col>)

alter table books

```

```
add unique(title)
```

## 20. Adding default constraint

```
alter table <table_name>
add constraint default(<value>) for <col>
```

```
alter table books
add constraint default(0) for pages
```

## 21. Adding a check constraint

```
alter table <table_name>
add constraint <constraint_name> <constraint>
```

```
alter table books
add constraint min_pages check(pages >= 0)
```

## 22. Dropping a named constraint

```
alter table <table_name>
drop <constraint_name>
```

```
alter table books
drop min_pages
```

## 23. Adding a foreign key constraint

```
alter table <table_name>
add foreign key (<fk_col>) references <other_table>(<other_col>)
```

```
alter table books
add foreign key (author) references authors(name)
```

## 24. Viewing the records of a table

```
select * from <table_name>
```

```
select * from books
go
```

## 25. Viewing only few columns of the records of a table

```
select <column_name>, ... from <table_name>
```

```
select isbn, price, type from books
go
```

## 26. Selecting different orders

```
select price, isbn, type from books
go
```

## 27. Finding the count

```
select count(*) from <table_name>
go
```

```
select count(*) from books
go
```

## 28. Finding the min and max of a given column

```
select min(<col_name>) from <table_name>
select max(<col_name>) from <table_name>
```

```
select min(price) from books
go
```

```
select max(price) from books
go
```

## 29. Finding the total

```
select sum(<col_name>) from <table_name>
```

```
select sum(price) from books
go
```

## 30. Selecting only the distinct

```
select distinct(<col_name>) from <table_name>
```

```
select distinct(author) from books
go
```

## 31. Selecting through alias

```
select <alias>.<col>, ... from <table> as <alias>
```

```
select t.name, t.id from topics as t
go
```

## 32. Selecting by order

```
select * from books order by <col_name> [asc|desc]
```

```
select * from books order by name
select * from books order by name desc
go
```

## 33. Restricting using where clause

```
select * from <table_name> where <col>=<value> [and|or] . . .
```

```
select * from books where name='C'
select * from books where subjects_id < 4
select id from books where name='C' and subjects_id < 4
go
```

## 34. Restricting using where clause and LIKE clause

```
select * from <table_name> where <col_name> like <expression>
```

```
% for any number of characters
- for single character
[] for range
^ for exclusivity
```

```
select * from books where name like '%Programming%'
select * from books where name like '%Engineering%'
select * from books where author like '_.%'
select * from books where name like '[abcd]'
select * from books where name like '%[^a]'
go
```

### 35. Restricting using where clause and IN clause

```
select * from <table> where <col> in (<value>, . . . )

select * from books where isbn in (123, 234, 523, 890)
go
```

### 36. Restricting using where clause and BETWEEN clause

```
select * from <table> where <col_name> between <min> and <max>

select * from books where isbn between 123 and 890
go
```

### 37. Updating records

```
update <table> set <col_name>=<value> where <expression>

update books set price = 10 where price < 10
go
```

### 38. Deleting records

```
delete from <table> where <expression>

delete from books where price < 0
go
```

### 39. Grouping the records

```
select <function> from <table> group by <column> [having <expression>]

select count(*) from books group by type
go

select count(*) from books group by type having price > 25
go
```

### 40. Cross joining tables

```
select * from <left_table> cross join <right_table>

select * from books cross join authors
```

```
go
```

#### 41. Inner Join between tables

```
select * from <left_table> join <right_table> on
<left_column>=<right_column>
```

```
select * from books join authors on author_name = name
go
```

#### 42. Outer Join between tables

```
select * from <left_table> left outer join <right_table> on
<left_column>=<right_column>
```

```
select * from <left_table> right outer join <right_table> on
<left_column>=<right_column>
```

```
select * from <left_table> full outer join <right_table> on
<left_column>=<right_column>
```

```
select * from books left outer join authors on author_name = name
go
```

```
select * from books right outer join authors on author_name = name
go
```

```
select * from books full outer join authors on author_name = name
go
```

#### 43. Adding trigger to a table

```
create trigger <name> on <table> [for|insteadof] [create|update|
delete] as <action>
```

#### 44. Listing the triggers on a table

```
sp_helptrigger[<table>]
```

#### 45. Dropping a trigger

```
drop trigger <name>
```

#### 46. Creating or updating a stored procedure

```
[create|alter] procedure <name>
[[@<param> <type> [=<default_value>]][output]]
. . .
as
<body>
```

#### 47. Creating or updating a function

```
[create | alter] function <name>
[[@<param> <type> [= <default_value>] ]
. . .
```

```
returns [table | <type> | [<variable> table <definition>]]  
as  
<body>
```

**48. Defining transactions**

```
begin transaction  
.  
.  
.  
if <condition>  
    rollback transaction  
.  
.  
.  
commit transaction
```

**49. Exporting the sql file**

```
osql [-E|-U <user_name>] [-P <password>] [-d <database>] [-i<file>]
```

```
osql -E -d glarimy -iglarimy.sql
```

## **Other QuickLook Titles from Glarimy**

A QUICK LOOK AT  
**JAVA THREADS**

First Edition

**Krishna Mohan Koyya**

**GLARIMY**

Glarimy QuickLook Series on Java Threads

1



# A Quick Look At Java 6 Fundamentals

## First Edition

Covers basic programming constructs of JDK 1.6

- Compiling and running Java programs
- Data Types, Variables, Operators
- Autoboxing
- Decision making
- Iterations
- Interacting with user
- Formatting Input and output
- Arrays
- Command Line Arguments
- String Handling
- 

With complete code illustrations.

The logo for Glarimy Technology Services, featuring the word "GLARIMY" in a stylized, bold, sans-serif font. The letters "G", "L", "A", "R", "I", and "M" are in a light grey color, while the letters "Y" and "Y" are in a teal color.

© 2011, Glarimy Technology Services