SQL Server QUICK GUIDE
FOR THE BEGINNING USER

LANGUAGE FUNDAMENTALS

Structured Query Language (SQL) is the method by which we are able to work with objects and their data inside our database. The SQL Server flavor of SQL is called Transact SQL or T-SQL for short. T-SQL contains a set of programming extensions that adds several features to basic SQL. The following list gives an overview of the T-SQL commands and their classification.

DDL

Data Definition Language: Commands that we use to create and alter object structures in the database. These commands are not targeted at changing the actual data. Each change is committed immediately and ends the transaction including all DML issued up to that point.

CREATE	Create a new object n the database.
ALTER	Change the structure of an existing object.
DROP	Remove an object from the database.

DML

Data Manipulation Language: Allows us to retrieve and make changes to the data in the database. Changes may be explicitly committed or rolled back.

SELECT	Query data in the database.
INSERT	Insert a new row into an existing table.
UPDATE	Change the value of existing row data in a table.
DELETE	Remove a row of data from an existing table.
TRUNCATE	Removes all rows from a table or view.
BULK INSERT	Inserts rows from a data file into a table or view.

DCL

Data Control Language: Allows us to control which users have privileges to access objects or carry out certain actions in the database.

GRANT	Give a role or privilege to a user.
REVOKE	Take a role or privilege away from a user.
DENY	Denies a specified permission to a security object, and prevents the object from inheriting permission through it's membership in a group or role.

Admin SQL

There are numerous commands used for administrative purposes. Below is a list of the basic commands used by DBAs to administer the databases and related objects.

BACKUP	Backs up an entire database, transaction log, or one or more files or filegroups.
RESTORE	Re-creates database and all of its associated files and then places them in their original location.
DROP	Remove an object from the database.
KILL	Terminates user process based on SPID- **use carefully.**
Exec sp_who;	Returns list of all current users.
SHUTDOWN	Shuts down entire instance.

BASIC SYNTAX AND EXAMPLES

Note: There are many variations and extensions to the syntax and examples provided below. See SQL Server Books Online for complete syntax diagrams and usage examples.

http://msdn2.microsoft.com/en-us/library/ms130214.aspx

DDL (Data Definition) WITH TABLES…

```
CREATE TABLE table_name(
<column definition><table constraint> [,…n])
```

```
CREATE TABLE MyCustomers(
CustID INT IDENTITY (100, 1)PRIMARY KEY,
CompanyName NVARCHAR2(50))
```

```
CREATE TABLE MyOrders (OrderID int, CustID int
REFERENCES MyCustomers(CustID))
```

```
ALTER TABLE table_name ALTER COLUMN column_name
 {DROP DEFAULT/SET DEFAULT/IDENTITY [(seed,
Increment)] ADD <column_definition> |
<table_constraint> DROP
<constraint_name/column_name>
```

```
ALTER TABLE MyCustomers ALTER COLUMN CustID
IDENTITY(200, 2)
```

```
ALTER TABLE Employee DROP COLUMN bday
```

```
ALTER TABLE MyCustomers ALTER COLUMN CompanyName
DROP DEFAULT
```

```
DROP TABLE names_cpy
```

DML (Data Manipulation) WITH TABLES…

SELECT is different than other DML statements in that it does not actually change/manipulate the data by itself. It is sometimes used with other commands to carry out DML using data retrieved from elsewhere in the database. Following is the basic query-only syntax.

```
SELECT col1, col2, …colx | * FROM table_name
[WHERE colx = expr] [ORDER BY colx];
```

```
SELECT fname, lname FROM staff
WHERE phone IS NOT NULL
ORDER BY lname;
```

True Data Manipulation…

```
INSERT INTO staff (fname, lname, ssn)
VALUES ('Chris', 'Plum', 318675309);
```

```
INSERT INTO staff SELECT * FROM new_hires;
```

```
DELETE FROM staff WHERE lname = 'Smith';
```

```
TRUNCATE TABLE Customers;
```

```
BULK INSERT AdventureWorks.Sales.SalesOrderDetail
FROM 'f:\orders\lineitem.tbl'
WITH (FIELDTERMINATOR = \t, ROWTERMINATOR = \n);
```

DCL (Data Control) … and an intro to roles and permissions.

Permissions are required to access the database. First a login must be created on the instance level. Then that login is used to create a database user. Then permissions must be assigned to the database. This is often done with the use of roles to simplify security administration. Roles are broken down into two categories: Server level roles and database level roles.

Fixed Server Roles	
bulkadmin	Members can run the BULK INSERT statement.
dbcreator	Members can create, alter, drop, and restore any database.
diskadmin	Members can manage disk files.
processadmin	Members can terminate processes that are running in an instance of SQL Server.
securityadmin	Members can manage logins and their properties.
serveradmin	Members can change server-wide configuration options and shut down the server.
setupadmin	Members can add remove linked servers, and also execute some system stored procedures.
sysadmin	Members can perform any activity in the server.

Fixed Database Roles	
db_accessadmin	Granted: ALTER ANY USER, CREATE SCHEMA
db_accessadmin	Granted with GRANT option: CONNECT
db_backupoperator	Granted: BACKUP DATABASE, BACKUP LOG, CHECKPOINT
db_datareader	Granted: SELECT
db_datawriter	Granted: DELETE, INSERT, UPDATE
db_owner	Granted with GRANT option: CONTROL Granted: VIEW ANY DATABASE
dbm_monitor	Granted: VIEW most recent status in Database Mirroring Monitor Important: The dbm_monitor fixed database role is created in the msdb database when the first database is registered in Database Mirroring Monitor. The new dbm_monitor role has no members until a system administrator assigns users to the role. Granted: VIEW ANY DATABASE
db_denydatareader	Denied: SELECT
db_denydatawriter	Denied: DELETE, INSERT, UPDATE
db_securityadmin	Granted: ALTER ANY APPLICATION ROLE, ALTER ANY ROLE, CREATE SCHEMA, VIEW DEFINITION Granted: VIEW ANY DATABASE

db_ddladmin	Granted: ALTER ANY ASSEMBLY, ALTER ANY ASYMMETRIC KEY, ALTER ANY CERTIFICATE, ALTER ANY CONTRACT, ALTER ANY DATABASE DDL TRIGGER, ALTER ANY DATABASE EVENT, NOTIFICATION, ALTER ANY DATASPACE, ALTER ANY FULL TEXT CATALOG, ALTER ANY MESSAGE TYPE, ALTER ANY REMOTE SERVICE BINDING, ALTER ANY ROUTE, ALTER ANY SCHEMA, ALTER ANY SERVICE, ALTER ANY SYMMETRIC KEY, CHECKPOINT, CREATE AGGREGATE, CREATE DEFAULT, CREATE FUNCTION, CREATE PROCEDURE, CREATE QUEUE, CREATE RULE, CREATE SYNONYM, CREATE TABLE, CREATE TYPE, CREATE VIEW, CREATE XML SCHEMA COLLECTION, REFERENCES
db_denydatareader	Denied: SELECT
db_denydatawriter	Denied: DELETE, INSERT, UPDATE
db_securityadmin	Granted: ALTER ANY APPLICATION ROLE, ALTER ANY ROLE, CREATE SCHEMA, VIEW DEFINITION Granted: VIEW ANY DATABASE

`GRANT permission [ ,...n ]`
`ON <object>`
`TO <database_principal>`
`[WITH GRANT OPTION]`
	`GRANT IMPERSONATE ON USER::HamithaL TO AccountsPayable17; GO`

Admin SQL

```
BACKUP DATABASE Customers TO DISK =
N'C:\DataBackup\Customers_db_200706041200.BAK'
```

```
RESTORE DATABASE Customers FROM DISK =
N'C:\DataBackup\Customers_db_200706041200.BAK'
```

```
BACKUP LOG Customers TO DISK =
N'C:\LogBackup\Customers_db_200706041200.TRN'
```

```
KILL 55;
```

```
(Find out who has current sessions on instance prior to
shutting down)
Exec sp_who;
GO
SHUTDOWN;
GO
```

CONCLUSION

There are many good online references for SQL statement execution. This document only bullet-points some of the very basic commands. It is a very powerful language that, when exploited, will enable you to produce very complex reports.

PERPETUAL TECHNOLOGIES, INC.

Perpetual Technologies, Inc. (PTI) provides mission-critical database and information systems support to commercial and government enterprises worldwide. Focused on improving performance and lowering costs, our subject-matter experts plan, design, develop, deploy, and manage SQL Server database environments to your exact specifications.

PERPETUAL TECHNOLOGIES, INC.

5649 Lee Road
Indianapolis, IN 46216
800-538-0453
www.PTI.net

PERPETUAL TECHNOLOGIES, INC.

SQL Server Quick Reference Guide

FOR THE BEGINNING USER

ADVANCING THE DATA-DRIVEN WORLD