Unit 06

INTERACTIVE SQL

UNIT STRUCTURE

6.0 Learning Objectives :

After learning this unit, you will be able to understand :

- Interactive SQL and its command file
- The various data types in SQL

6.1 Introduction :

Interactive SQL (dbisql) is a utility for entering SQL statements. If you use Interactive SQL to work with your database schema, instead of executing the SQL statements one at a time, build up the set of commands in a dbisql command file. Then you can execute this file in dbisql to build the database.

The definitions of the database objects form the database schema. You can think of the schemaas an empty database. The SQL statements for creating and modifying schemas are called the data definition language (DDL).

Note : Only one user at a time can perform DDL statements on a table. IQ locks a table during DDL operations on it. Users may, however, perform DDL on other objects in the same database at the same time.

If you use a tool other than Interactive SQL, all the information in these topics concerning SQL statements still applies.

Interactive SQL Command File

An Interactive SQL command file is a text file with semicolons placed at the end of commands as shown below.

```
CREATE TABLE t1 (..);
CREATE TABLE t2 (..);
CREATE LF INDEX i2 ON t2 (..);
..
```

An Interactive SQL command file usually carries the extension .sql. To execute a command file, either paste the contents of the file into the Interactive

SQL command window (if the file has less than 500 lines) or enter a command that reads the file into the command window. For example, the READ statement:

readmakedb

Reads the Interactive SQL commands in the file makedb.sql.

6.2 Data Types :

SQL data type is an attribute that specifies type of data of any object. Each column, variable and expression has related data type in SQL. You would use these data types while creating your tables. You would choose a particular data type for a table column based on your requirement. SQL Server offers six categories of data types for your use :

Exact Numeric Data Types :

DATA TYPE	FROM	TO
Bigint	–9,223,372,036,854,775,808	9,223,372,036,854,775,807
Int	–2,147,483,648	2,147,483,647
Smallint	–32,768	32,767
Tinyint	0	255
Bit	0	1
Decimal	–10^38 +1	10^38 –1
Numeric	–10^38 +1	10^38 –1
Money	–922,337,203,685,477.5808	+922,337,203,685,477.5807
Smallmoney	–214,748.3648	+214,748.3647

Approximate Numeric Data Types :

DATA TYPE	FROM	TO
Float	–1.79E + 308	1.79E + 308
Real	–3.40E + 38	3.40E + 38

Date and Time Data Types :

DATA TYPE	FROM	TO
Datetime	Jan 1, 1753	Dec 31, 9999
Smalldatetime	Jan 1, 1900	Jun 6, 2079
Date	Stores a date like June 30, 1991	
Time	Stores a time of day like 12:30 P.M.	

Note : Here, date time has 3.33 milli seconds accuracy where as small date time has 1 minute accuracy.

Character Strings Data Types :

DATA TYPE	FROM	TO
Char	Char	Maximum length of 8,000 characters. (Fixed length non–Unicode characters)
Varchar	Varchar	Maximum of 8,000 characters. (Variable–length non–Unicode data).
varchar(max)	varchar(max)	Maximum length of 231 characters, Variable–length non–Unicode data (SQL Server 2005 only).
Text	text	Variable–length non–Unicode data with a maximum length of 2,147,483,647 characters.

Unicode Character Strings Data Types :

DATA TYPE	Description
Nchar	Maximum length of 4,000 characters. (Fixed length Unicode)
Nvarchar	Maximum length of 4,000 characters. (Variable length Unicode)
nvarchar(max)	Maximum length of 231characters (SQL Server 2005 only). (Variable length Unicode)
Ntext	Maximum length of 1,073,741,823 characters. (Variable length Unicode)

Binary Data Types :

DATA TYPE	Description
Binary	Maximum length of 8,000 bytes (Fixed–length binary data)
Varbinary	Maximum length of 8,000 bytes. (Variable length binary data)
varbinary(max)	Maximum length of 231 bytes (SQL Server 2005 only). (Variable length Binary data)
Image	Maximum length of 2,147,483,647 bytes. (Variable length Binary Data)

Misc Data Types :

DATA TYPE	Description
sql_variant	Stores values of various SQL Server–supported data types, except text, ntext, and timestamp.
Timestamp	Stores a database–wide unique number that gets updated every time a row gets updated
uniqueidentifier	Stores a globally unique identifier (GUID)
xml	Stores XML data. You can store xml instances in a column or a variable (SQL Server 2005 only).
cursor	Reference to a cursor object
table	Stores a result set for later processing