CIS560

Data Modification

Part 1

1

Data Modification

- Remember, SQL has two parts:
  - Data Manipulation Language (DML)
  - Data Definition Language (DDL)
- We have been learning the SELECT statement.
- SELECT is one of six statements in DML.
- DML allows
  - Inserting (INSERT statement)
  - Updating (UPDATE statement)
  - Deleting (DELETE and TRUNCATE statements)
  - Combinations (MERGE statement)

2

Inserting Data

- There are several ways to insert data in SQL.
- We'll just look at three (all standard):
  - INSERT ... VALUES …
  - INSERT … SELECT …
  - INSERT … EXEC …

3

Inserting Data

- All begin with similar syntax:

```
INSERT [INTO] Table(<column list>)
```

  - INTO is optional.
  - Column list is optional, but always include it!
- If a column in the table is omitted…
  - But it has a default constraint, a default value is assigned.
  - Otherwise, if it allows nulls, then a NULL is inserted.
  - Otherwise, the INSERT fails.

4

Inserting Data

- Sometimes we need the value from identity column.
- There are three functions:
  - `@@IDENTITY`
    Returns the last identity value generated in the session.
  - `SCOPE_IDENTITY()`
    Returns the last identity value generated by the session in the current scope.
  - `IDENT_CURRENT(<table name>)`
    Returns the last identity value generated globally.

Inserting Data

- IDENTITY is non-standard, but simple.
- Standard SQL defines a SEQUENCE object.
- Syntax

```
CREATE SEQUENCE <schema>.<sequence name> AS INT
   MINVALUE 1
   START WITH 1
   INCREMENT BY 1
   NO CYLE;
```

- Other options exist too, but more advanced.

Syntax

```
INSERT [INTO] Table(<column list>)
VALUES
   ({DEFAULT | NULL | expression} [ ,...n ]) [ ,...];


INSERT [INTO] Table(<column list>)
<SELECT statement>;


INSERT [INTO] Table(<column list>)
{ EXEC | EXECUTE }
   { <stored_procedure_invocation>
      | <string_variable>
      | [ N ]'tsql_string' };
```