

Introduction

This paper will discuss the basic processing of database data, including SQL statements, the use of transactions, and the use of Identity and related Identity functions for the purpose of autonumbering.

Working with data

Four basic operations may be performed on data stored in a database. These are commonly referred to as CRUD operations which stands for create, read, update and delete. The SQL statements used to perform each operation follow a similar format with nuanced differences.

Create

The `INSERT INTO` statement is used when creating one or more new records in a database. When creating a record, the table where the data is being added must be defined. In order to reduce the risk of unexpected errors, it is recommended that the columns for which data will be entered be explicitly defined. Once the table and columns to be inserted have been defined, one or more row constructors may be added to the SQL statement; the row constructors contain the data values being added to the database.

Example ¹

```
-- Basic insert statement example adding two records to a database
INSERT INTO table_name (column1, column2, column3)
VALUES (value1_1, value1_2, value1_3),
       (value2_1, value2_2, value2_3);
```

Read

To read data from a database the `SELECT` statement is used. In its simplest form, the select statement will return all the rows and columns of a table. A `WHERE` clause may be added to filter out unwanted results, and columns may be defined to retrieve only specified columns.

Example ²

```
-- Basic select statement retrieving all rows and columns of a table
SELECT * FROM table_name;
-- Basic select statement retrieving specific columns from a table
SELECT column1, column2 FROM table_name;
```

Update

Modification of a record that is already held within a database is done using the `UPDATE` statement. While similar to the select statement, the column or columns being modified must be defined. It is essential to remember a `WHERE` clause when updating data, or all the records in a table will have the specified columns updated.

Example ³

```
-- Basic update statment with where clause
UPDATE table_name
SET column1 = value1, column2 = value2
WHERE condition;
```

Delete

Removing a record from a database is relatively straightforward with the use of the `DELETE` statement. Like the update statement, it is important to remember a `WHERE` clause; otherwise, all the data in a table will be removed.

Example ⁴

```
-- Basic delete statement with where clause
DELETE FROM table_name WHERE condition;
```

SQL Transactions

A simple description of a transaction is any interaction with a database, whether retrieving or modifying data. A more detailed description is that a transaction is a unit of work containing one or more SQL statements that may be committed to the database or rolled back prior to committing ⁵.

An example of a transaction that contains multiple SQL statements is transferring money from a checking to a savings account:

1. `BEGIN TRANSACTION`
2. `UPDATE` savings account balance to remove the amount being transferred
3. `UPDATE` the checking account balance to add the amount being transferred
4. `COMMIT TRANSACTION`

These four steps may be considered one unit of work. This is because if either step two or three were to fail, the user would not want the transfer to be processed as money could go missing or appear out of nowhere. Since the steps one through four are part of the same a transaction, if any step were to fail, a rollback may be performed prior to any changes being recored in the database. If all steps of the transaction were successful, the changes to the database would all be made at once.

Auto numbering with Identity

In MS SQL, auto numbering is done using the Identity column definition. The most common use for using Identity would be to automatically assign a unique integer value to a table row as a primary key. By default, when using Identity, it will start at one and increment by one. However, this may be changed by defining a seed and increment value (ex: `IDENTITY(12, 2)` auto numbering will start at 12 and increment by 2). It is important to note that a table may only have one Identity column defined.

When writing a series of SQL statements, it might be necessary to retrieve the last used identity value. This can be done using system functions `@@IDENTITY` or `IDENT_CURRENT()`. The `@@IDENTITY` function does not take any parameters and will return the identity value of the last insert statement. If the last insert statement did not affect any rows, a value of `NULL` would be returned. Caution must be used when using this function as the value returned may not be from a table that is different than expected as it returns the last identity value from any table.

By using the `IDENT_CURRENT()` function, unexpected results may be eliminated by passing in the name of the table of interest as a parameter. For example, if the user wants to retrieve the last identity value used on the table 'tblContacts' then the function would be called by using `IDENT_CURRENT('tblContacts')`. There are a few cases where the `IDENT_CURRENT()` function will return a value of `NULL`, this will occur if the table in question does not contain any data or the user does not have the required permissions to access a table.

Summary

Manipulation of large amounts of data within a SQL database can be easily accomplished with a few simple statements. It is important to remember to include appropriate filtering with the use of a where clause when necessary in order to prevent a data mishap and maintain data integrity. Data integrity may also be maintained through the effective use of transactions and their ability to be rolled back before committing a change to the database. Ensuring that all records in a table have a unique primary key is essential to easily access the correct records. This task is made easy by using Identity and its associated functions.

1. SQL INSERT INTO Statement. (2022). W3Schools. https://www.w3schools.com/sql/sql_insert.asp ↗

2. SQL SELECT Statement. (2022). W3Schools. https://www.w3schools.com/sql/sql_select.asp ↗

3. SQL UPDATE Statement. (2022). W3Schools. https://www.w3schools.com/sql/sql_update.asp ↗

4. SQL DELETE Statement. (2022). W3Schools. https://www.w3schools.com/sql/sql_delete.asp ↗

5. Transaction Management. (2022). Oracle Help Center. [https://docs.oracle.com/cd/B19306_01/server.102/b14220/transact.htm#%7E:text=A%20transaction%20is%20a%20logical,\(undone%20from%20the%20database\).](https://docs.oracle.com/cd/B19306_01/server.102/b14220/transact.htm#%7E:text=A%20transaction%20is%20a%20logical,(undone%20from%20the%20database).) ↗