Introduction to SQL

Insert, Update, and Delete

Jon Flanders
@jonflanders

pluralsight
hardcore developer training

What you will learn...

What you will learn...

- How to add, change, and delete data inside of Tables

CRUD

CRUD

- Create

CRUD

- **Create**
  - Add new rows to a Table

CRUD

- **Create**
  - Add new rows to a Table
- **Read**

CRUD

- **Create**
  - Add new rows to a Table
- **Read**
  - We've already done this!

CRUD

- **Create**
  - Add new rows to a Table
- **Read**
  - We’ve already done this!
- **Update**

CRUD

- **Create**
    - Add new rows to a Table
- **Read**
    - We've already done this!
- **Update**
    - Change a value in one or more of the columns in one or more rows in a Table

CRUD

- **Create**
  - Add new rows to a Table
- **Read**
  - We’ve already done this!
- **Update**
  - Change a value in one or more of the columns in one or more rows in a Table
- **Delete**

CRUD

- **Create**
    - Add new rows to a Table
- **Read**
    - We’ve already done this!
- **Update**
    - Change a value in one or more of the columns in one or more rows in a Table
- **Delete**
    - Remove one or more rows from a Table

CRUD

- **Create**
  - Add new rows to a Table
- **Read**
  - We've already done this!
- **Update**
  - Change a value in one or more of the columns in one or more rows in a Table
- **Delete**
  - Remove one or more rows from a Table
- **Should really be ISUD**

CRUD

- **Create**
  - Add new rows to a Table
- **Read**
  - We’ve already done this!
- **Update**
  - Change a value in one or more of the columns in one or more rows in a Table
- **Delete**
  - Remove one or more rows from a Table
- **Should really be ISUD**
  - INSERT, SELECT, UPDATE, and DELETE are the actual commands

CRUD

- **Create**
  - Add new rows to a Table
- **Read**
  - We’ve already done this!
- **Update**
  - Change a value in one or more of the columns in one or more rows in a Table
- **Delete**
  - Remove one or more rows from a Table
- **Should really be ISUD**
  - INSERT, SELECT, UPDATE, and DELETE are the actual commands
  - I guess CRUD sounds better

INSERT

INSERT

- **The INSERT command enables you to put new data into a Table**

INSERT

- The INSERT command enables you to put new data into a Table
- Really should be known as "INSERT INTO"

INSERT

- **The INSERT command enables you to put new data into a Table**
- **Really should be known as "INSERT INTO"**
    - INTO is a required part of the command

INSERT

- **The INSERT command enables you to put new data into a Table**
- **Really should be known as "INSERT INTO"**
  - INTO is a required part of the command
- **Table name follows INSERT INTO**

INSERT

- **The INSERT command enables you to put new data into a Table**
- **Really should be known as "INSERT INTO"**
  - INTO is a required part of the command
- **Table name follows INSERT INTO**
  - Only one Table at a time is allowed

INSERT

- **The INSERT command enables you to put new data into a Table**
- **Really should be known as “INSERT INTO”**
  - INTO is a required part of the command
- **Table name follows INSERT INTO**
  - Only one Table at a time is allowed
- **The list of columns follows in a set of parens**

INSERT

- **The INSERT command enables you to put new data into a Table**
- **Really should be known as "INSERT INTO"**
  - INTO is a required part of the command
- **Table name follows INSERT INTO**
  - Only one Table at a time is allowed
- **The list of columns follows in a set of parens**
  - Separated by commas

INSERT

- **The INSERT command enables you to put new data into a Table**
- **Really should be known as "INSERT INTO"**
  - INTO is a required part of the command
- **Table name follows INSERT INTO**
  - Only one Table at a time is allowed
- **The list of columns follows in a set of parens**
  - Separated by commas
  - All required columns must be present

INSERT

- **The INSERT command enables you to put new data into a Table**
- **Really should be known as "INSERT INTO"**
  - INTO is a required part of the command
- **Table name follows INSERT INTO**
  - Only one Table at a time is allowed
- **The list of columns follows in a set of parens**
  - Separated by commas
  - All required columns must be present
- **After the columns comes the VALUES keyword**

INSERT

- **The INSERT command enables you to put new data into a Table**
- **Really should be known as "INSERT INTO"**
  - INTO is a required part of the command
- **Table name follows INSERT INTO**
  - Only one Table at a time is allowed
- **The list of columns follows in a set of parens**
  - Separated by commas
  - All required columns must be present
- **After the columns comes the VALUES keyword**
  - Followed by a parens that hold onto the list of values to be put into each column

INSERT

- **The INSERT command enables you to put new data into a Table**
- **Really should be known as “INSERT INTO”**
  - INTO is a required part of the command
- **Table name follows INSERT INTO**
  - Only one Table at a time is allowed
- **The list of columns follows in a set of parens**
  - Separated by commas
  - All required columns must be present
- **After the columns comes the VALUES keyword**
  - Followed by a parens that hold onto the list of values to be put into each column
  - Each column in the list must have a matching value in the value list

INSERT INTO

INSERT INTO

```
INSERT INTO person
(person_id, person_first_name, person_last_name,
person_contacted_number,person_date_last_contacted,
person_date_added )
VALUES (1, 'Jon', 'Flanders', 5,'2013-09-14
11:43:31','2013-01-14 11:43:31');
```

Inserting Multiple Rows

Inserting Multiple Rows

- You can insert multiple rows using multiple value lists

Inserting Multiple Rows

- **You can insert multiple rows using multiple value lists**
  - Separate each value list with a comma

Inserting Multiple Rows

- **You can insert multiple rows using multiple value lists**
  - Separate each value list with a comma
- **You can also use a query to select multiple rows “into” another Table**

Inserting Multiple Rows

- **You can insert multiple rows using multiple value lists**
  - Separate each value list with a comma
- **You can also use a query to select multiple rows “into” another Table**

```
INSERT INTO person
(person_id, person_first_name, person_last_name,
person_contacted_number,person_date_last_contact
ed,person_date_added )
VALUES (1, 'Jon', 'Flanders', 5,'2013-09-14
11:43:31','2013-01-14 11:43:31'),
(2, 'Shannon', 'Ahern', 0,'2013-08-14
11:43:31','2013-02-14 11:43:31');
```

Inserting Multiple Rows

- **You can insert multiple rows using multiple value lists**
  - Separate each value list with a comma
- **You can also use a query to select multiple rows “into” another Table**

```
INSERT INTO person
(person_id, person_first_name, person_last_name,
person_contacted_number,person_date_last_contact
ed,person_date_added )
VALUES (1, 'Jon', 'Flanders', 5,'2013-09-14
11:43:31','2013-01-14 11:43:31'),
(2, 'Shannon', 'Ahern', 0,'2013-08-14
11:43:31','2013-02-14 11:43:31');
```

```
INSERT INTO person p
SELECT * from old_person op
WHERE op.person_id > 300;
```

UPDATE

UPDATE

- UPDATE enables you to change one or more columns in a Table

UPDATE

- UPDATE enables you to change one or more columns in a Table
- Without a WHERE clause, UPDATE (like DELETE) will affect all the rows in the Table

UPDATE

- UPDATE enables you to change one or more columns in a Table
- Without a WHERE clause, UPDATE (like DELETE) will affect all the rows in the Table
- SET keyword follows Table name (and alias)

UPDATE

- UPDATE enables you to change one or more columns in a Table
- Without a WHERE clause, UPDATE (like DELETE) will affect all the rows in the Table
- SET keyword follows Table name (and alias)

```
        UPDATE email_address e
   SET e.email_address_person_id = 5
WHERE e.email_address = ‘aaron@mail.com’;
```

DELETE

DELETE

- **DELETE deletes one or more rows**

DELETE

- **DELETE deletes one or more rows**
  - Permanently!

DELETE

- **DELETE deletes one or more rows**
  - Permanently!
  - Be careful!

DELETE

- **DELETE deletes one or more rows**
  - Permanently!
  - Be careful!
- **Command is actually "DELETE FROM"**

DELETE

- **DELETE deletes one or more rows**
  - Permanently!
  - Be careful!
- **Command is actually "DELETE FROM"**
- **Table name comes after the FROM**

DELETE

- **DELETE deletes one or more rows**
  - Permanently!
  - Be careful!
- **Command is actually “DELETE FROM”**
- **Table name comes after the FROM**
- **Only one Table at a time allowed**

DELETE

- **DELETE deletes one or more rows**
  - Permanently!
  - Be careful!
- **Command is actually "DELETE FROM"**
- **Table name comes after the FROM**
- **Only one Table at a time allowed**
- **With no conditions – DELETE FROM deletes all rows!**

DELETE

- **DELETE deletes one or more rows**
  - Permanently!
  - Be careful!
- **Command is actually "DELETE FROM"**
- **Table name comes after the FROM**
- **Only one Table at a time allowed**
- **With no conditions – DELETE FROM deletes all rows!**

```
DELETE FROM person;
```

Transactions

Transactions

- A Transaction is a specific computer science concept

Transactions

- **A Transaction is a specific computer science concept**
    - Not the same as a ATM or bank transaction

Transactions

- **A Transaction is a specific computer science concept**
    - Not the same as a ATM or bank transaction
- **A Transaction is a construct that creates a “context” around one or more SQL statements**

ACID

ACID

- Another common SQL-related acronym

ACID

- Another common SQL-related acronym
- It describes the qualities of a Transaction

ACID

- Another common SQL-related acronym
- It describes the qualities of a Transaction
- Atomic

ACID

- **Another common SQL-related acronym**
- **It describes the qualities of a Transaction**
- **Atomic**
    - The Transaction either happens or it doesn’t

ACID

- **Another common SQL-related acronym**
- **It describes the qualities of a Transaction**
- **Atomic**
  - The Transaction either happens or it doesn’t
- **Consistent**

ACID

- **Another common SQL-related acronym**
- **It describes the qualities of a Transaction**
- **Atomic**
    - The Transaction either happens or it doesn't
- **Consistent**
    - The Transaction leaves the database in a consistent state

ACID

- **Another common SQL-related acronym**
- **It describes the qualities of a Transaction**
- **Atomic**
  - The Transaction either happens or it doesn’t
- **Consistent**
  - The Transaction leaves the database in a consistent state
- **Isolated**

ACID

- **Another common SQL-related acronym**
- **It describes the qualities of a Transaction**
- **Atomic**
  - The Transaction either happens or it doesn't
- **Consistent**
  - The Transaction leaves the database in a consistent state
- **Isolated**
  - The Transaction happens in a isolated (serial) way

ACID

- **Another common SQL-related acronym**
- **It describes the qualities of a Transaction**
- **Atomic**
    - The Transaction either happens or it doesn't
- **Consistent**
    - The Transaction leaves the database in a consistent state
- **Isolated**
    - The Transaction happens in a isolated (serial) way
- **Durable**

ACID

- **Another common SQL-related acronym**
- **It describes the qualities of a Transaction**
- **Atomic**
    - The Transaction either happens or it doesn’t
- **Consistent**
    - The Transaction leaves the database in a consistent state
- **Isolated**
    - The Transaction happens in a isolated (serial) way
- **Durable**
    - The Transaction will be stored permanently, even in the face of a disaster (e.g. power outage)

Transaction Syntax

Transaction Syntax

```
START TRANSACTION;
DELETE FROM
person;
COMMIT;
-- or ROLLBACK;
```

DELETE + WHERE

DELETE + WHERE

- You can limit the number of rows that are deleted

DELETE + WHERE

- **You can limit the number of rows that are deleted**
  - Use the same technique as limiting the number of rows returned from a SELECT statement

DELETE + WHERE

- **You can limit the number of rows that are deleted**
    - Use the same technique as limiting the number of rows returned from a SELECT statement
- **The WHERE clause comes after the Table name**

DELETE + WHERE

- **You can limit the number of rows that are deleted**
  - Use the same technique as limiting the number of rows returned from a SELECT statement
- **The WHERE clause comes after the Table name**

```
DELETE FROM person p
WHERE p.person_first_name LIKE ‘J%’;
```

DELETE + WHERE + IN

DELETE + WHERE + IN

- The IN clause can also be used after the WHERE clause to list a set of rows to be deleted

DELETE + WHERE + IN

- **The IN clause can also be used after the WHERE clause to list a set of rows to be deleted**
    - Often useful for deleting a large number of rows based on the primary key

DELETE + WHERE + IN

- **The IN clause can also be used after the WHERE clause to list a set of rows to be deleted**
  - Often useful for deleting a large number of rows based on the primary key

```
      DELETE FROM person p
WHERE p.person_id IN (1,2,3,4,5,6);
```

Summary

Summary

- INSERT, UPDATE, and DELETE are the three SQL commands you want to learn if you are going to be modifying data in your database tables

Summary

- INSERT, UPDATE, and DELETE are the three SQL commands you want to learn if you are going to be modifying data in your database tables
- Remember to learn transactions in full to be able to protect your SQL statements against incorrect behavior