```
mysql> SHOW STATUS LIKE '%trans_commit_count%';
+----------------------------------+-------+
| Variable_name                    | Value |
+----------------------------------+-------+
| Ndb_api_trans_commit_count_session | 1   |
| Ndb_api_trans_commit_count_slave   | 0   |
| Ndb_api_trans_commit_count         | 2   |
+----------------------------------+-------+
3 rows in set (0.00 sec)
```

From this you can determine that 1 transaction has been committed in the current `mysql` client session, and 2 transactions have been committed on this `mysqld` since it was last restarted.

You can see how various NDB API counters are incremented by a given SQL statement by comparing the values of the corresponding `_session` status variables immediately before and after performing the statement. In this example, after getting the initial values from SHOW STATUS, we create in the `test` database an NDB table, named `t`, that has a single column:

```
mysql> SHOW STATUS LIKE 'ndb_api%session%';
+------------------------------------------+--------+
| Variable_name                            | Value  |
+------------------------------------------+--------+
| Ndb_api_wait_exec_complete_count_session | 2      |
| Ndb_api_wait_scan_result_count_session   | 0      |
| Ndb_api_wait_meta_request_count_session  | 3      |
| Ndb_api_wait_nanos_count_session         | 820705 |
| Ndb_api_bytes_sent_count_session         | 132    |
| Ndb_api_bytes_received_count_session     | 372    |
| Ndb_api_trans_start_count_session        | 1      |
| Ndb_api_trans_commit_count_session       | 1      |
| Ndb_api_trans_abort_count_session        | 0      |
| Ndb_api_trans_close_count_session        | 1      |
| Ndb_api_pk_op_count_session              | 1      |
| Ndb_api_uk_op_count_session              | 0      |
| Ndb_api_table_scan_count_session         | 0      |
| Ndb_api_range_scan_count_session         | 0      |
| Ndb_api_pruned_scan_count_session        | 0      |
| Ndb_api_scan_batch_count_session         | 0      |
| Ndb_api_read_row_count_session           | 1      |
| Ndb_api_trans_local_read_row_count_session | 1    |
+------------------------------------------+--------+
18 rows in set (0.00 sec)

mysql> USE test;
Database changed
mysql> CREATE TABLE t (c INT) ENGINE NDBCLUSTER;
Query OK, 0 rows affected (0.85 sec)
```

Now you can execute a new SHOW STATUS statement and observe the changes, as shown here (with the changed rows highlighted in the output):

```
mysql> SHOW STATUS LIKE 'ndb_api%session%';
+------------------------------------------+-----------+
| Variable_name                            | Value     |
+------------------------------------------+-----------+
| Ndb_api_wait_exec_complete_count_session | 8         |
| Ndb_api_wait_scan_result_count_session   | 0         |
| Ndb_api_wait_meta_request_count_session  | 17        |
| Ndb_api_wait_nanos_count_session         | 706871709 |
| Ndb_api_bytes_sent_count_session         | 2376      |
| Ndb_api_bytes_received_count_session     | 3844      |
| Ndb_api_trans_start_count_session        | 4         |
| Ndb_api_trans_commit_count_session       | 4         |
| Ndb_api_trans_abort_count_session        | 0         |
| Ndb_api_trans_close_count_session        | 4         |
| Ndb_api_pk_op_count_session              | 6         |
```