

SLONIK SCRIPT

PREAMBLE

```
CLUSTER NAME=ident;  
NODE ival ADMIN CONNINFO='string';
```

STATEMENT

```
TRY {statements} [ ON SUCCESS {statements} ] [ ON ERROR {statements} ]  
ECHO 'string';  
EXIT int;
```

CLUSTER and NODE

```
INIT CLUSTER ([ID=ival][, COMMENT='string']);  
STORE NODE (ID=ival, COMMENT='string', SPOOLNODE=boolean)[, EVENT NODE=ival];  
DROP NODE (ID=ival, EVENT NODE=ival);  
UNINSTALL NODE (ID=ival);  
RESTART NODE (ID=ival);
```

PATH and LISTEN

```
STORE PATH (SERVER=ival, CLIENT=ival, CONNINFO='string', CONNRETRY=ival);  
DROP PATH (SERVER=ival, CLIENT=ival, EVENT NODE=ival);  
STORE LISTEN (ORIGIN=ival, PROVIDER=ival, RECEIVER=ival);  
DROP LISTEN (ORIGIN=ival, PROVIDER=ival, RECEIVER=ival);
```

SET

```
CREATE SET (ID=ival, ORIGIN=ival, COMMENT='string');  
DROP SET (ID=ival, ORIGIN=ival);  
MERGE SET (ID=ival, ADD ID=ival, ORIGIN=ival);  
SUBSCRIBE SET (ID=ival, PROVIDER=ival, RECEIVER=ival, FORWARD=boolean);  
UNSUBSCRIBE SET (ID=ival, RECEIVER=ival);  
LOCK SET (ID=ival, ORIGIN=ival);  
UNLOCK SET (ID=ival, ORIGIN=ival);  
MOVE SET (ID=ival, OLD ORIGIN=ival, NEW ORIGIN=ival);
```

TABLE, SEQUENCE, TRIGGER

```
TABLE ADD KEY (NODE ID=ival, FULLY QUALIFIED NAME='string');
SET ADD TABLE (SET ID=ival, ORIGIN=ival, ID=ival, FULLY QUALIFIED NAME='string',
KEY={'string' | SERIAL}, COMMENT='string');
SET ADD SEQUENCE (SET ID=ival, ORIGIN=ival, ID=ival, FULLY QUALIFIED NAME='string',
COMMENT='string');
SET DROP TABLE (ORIGIN=ival, ID=ival);
SET DROP SEQUENCE (ORIGIN=ival, ID=ival);
SET MOVE TABLE (ORIGIN=ival, ID=ival, NEW SET=ival);
SET MOVE SEQUENCE (ORIGIN=ival, ID=ival, NEW SET=ival);
STORE TRIGGER (TABLE ID=ival, TRIGGER NAME='string'[, EVENT NODE=ival]);
DROP TRIGGER (TABLE ID=ival, TRIGGER NAME='string'[, EVENT NODE=ival]);
```

OTHERS

```
FAILOVER (ID=ival, BACKUP NODE=ival);
EXECUTE SCRIPT (SET ID=ival, FILENAME='string'[, EVENT NODE=ival][, EXECUTE ONLY
ON=ival]);
UPDATE FUNCTIONS (ID=ival);
WAIT FOR EVENT (ORIGIN={ival | ALL}, CONFIRMED={ival | ALL}, WAIT ON=ival, TIMEOUT=ival);
```

IDIOM

SWITCH OVER

```
LOCK SET (ID=1, ORIGIN=1);
WAIT FOR EVENT (ORIGIN=1, CONFIRMED=2);
MOVE SET (ID=1, OLD ORIGIN=1, NEW ORIGIN=2);
WAIT FOR EVENT (ORIGIN=1, CONFIRMED=2);
```