SQL

DELETE STATEMENT

The DELETE Statement

the DELETE statement

removes records from a database


```
DELETE FROM table_name
WHERE conditions;
```

FOREIGN KEY Constraint

ON DELETE CASCADE

if a specific value *from the parent table's primary key* has been deleted, all the records *from the child table* referring to this value will be removed as well

DROP vs TURNCATE vs DELETE

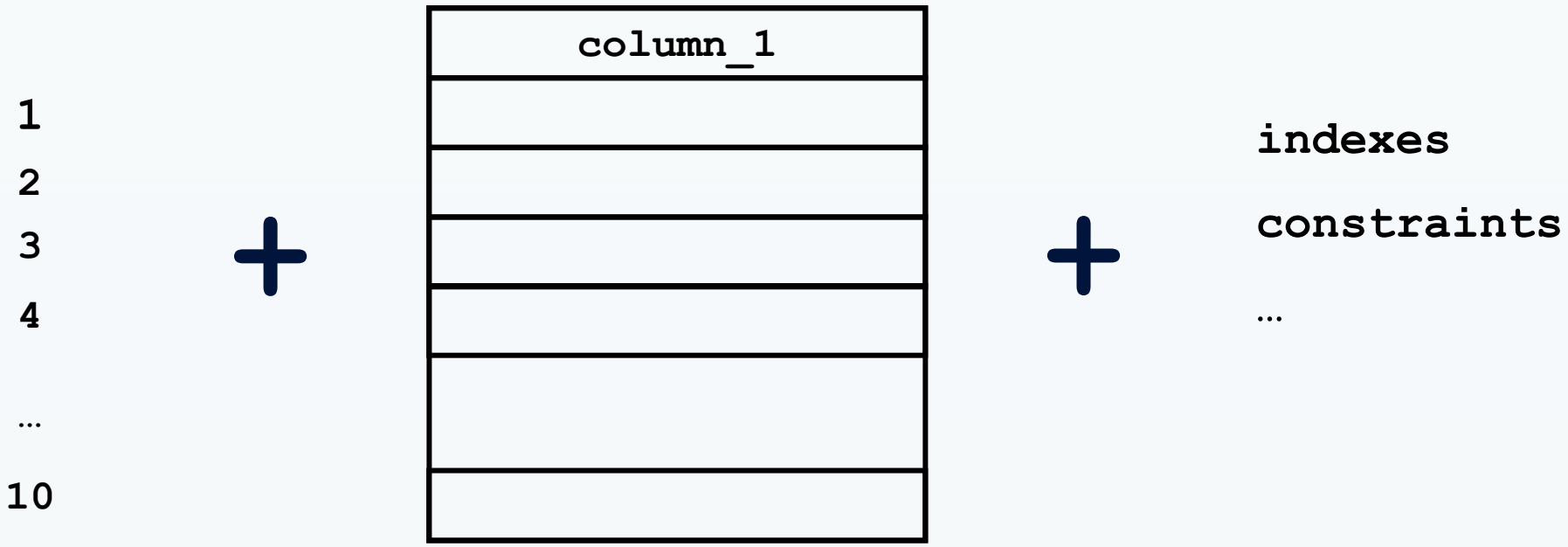
DROP vs TRUNCATE vs DELETE

DROP

column_1
1
2
3
4
…
10

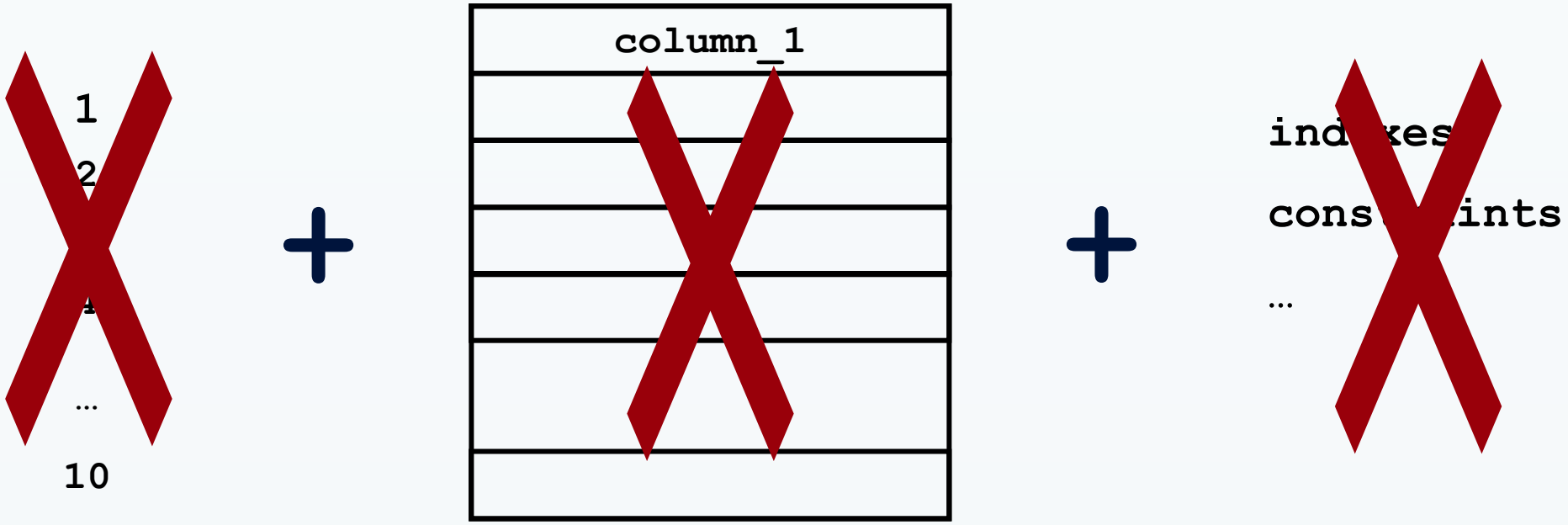
DROP vs TRUNCATE vs DELETE

DROP



DROP vs TRUNCATE vs DELETE

DROP



DROP vs TRUNCATE vs DELETE

DROP

- you won't be able to roll back to its initial state, or to the last COMMIT statement

use DROP TABLE only when you are sure you aren't going to use the table in question anymore

DROP vs TRUNCATE vs DELETE

TRUNCATE

column_1
1
2
3
4
…
10

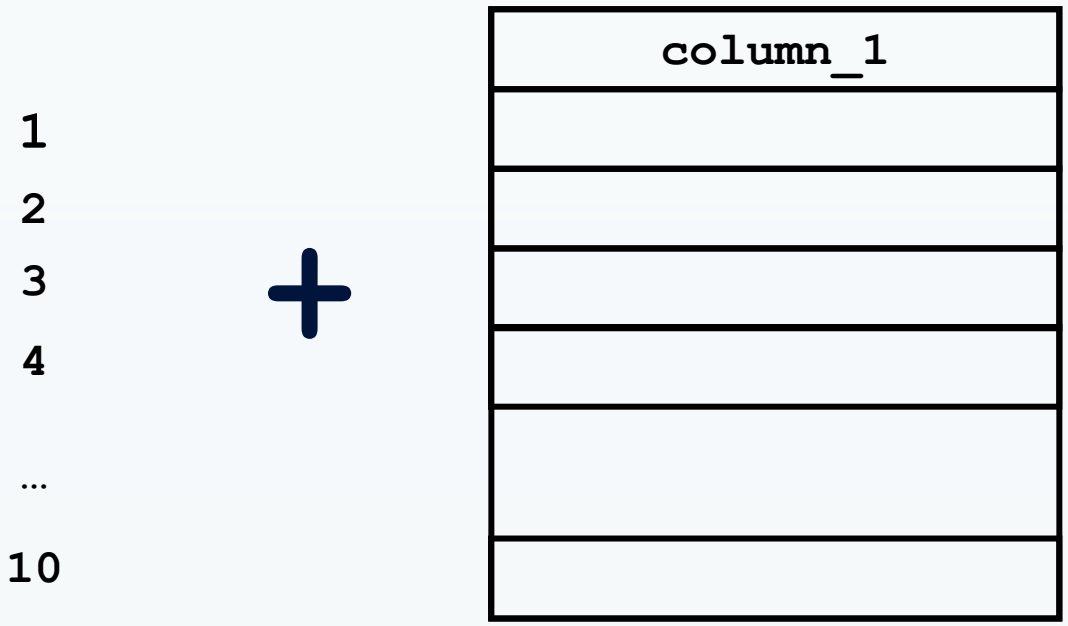
DROP vs TRUNCATE vs DELETE

TRUNCATE ~ DELETE without WHERE

column_1
1
2
3
4
…
10

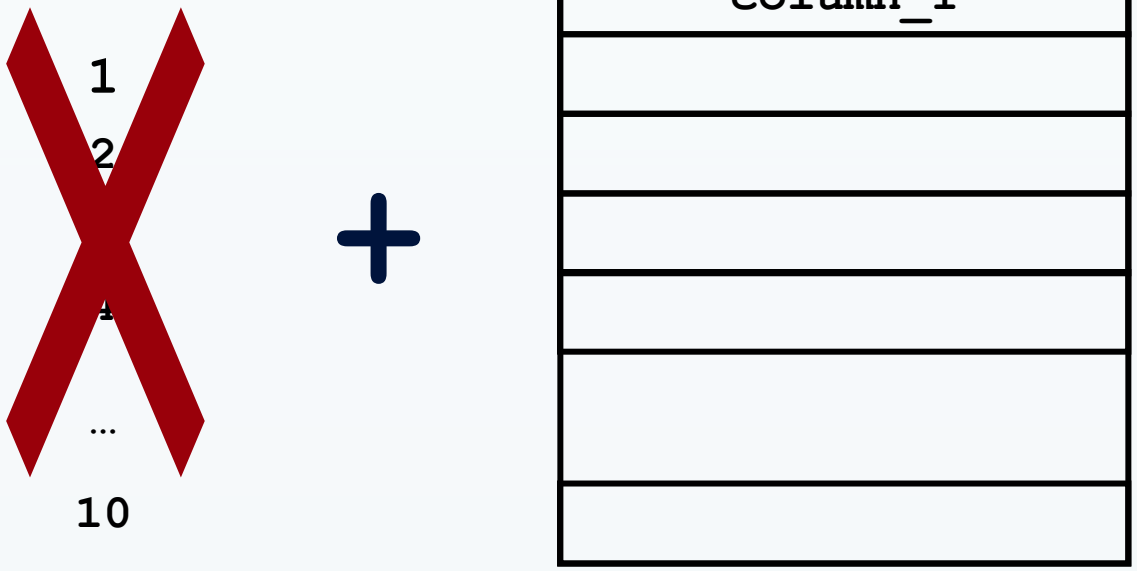
DROP vs TRUNCATE vs DELETE

TRUNCATE ~ DELETE without WHERE



DROP vs TRUNCATE vs DELETE

TRUNCATE ~ DELETE without WHERE



DROP vs TRUNCATE vs DELETE

TRUNCATE

when truncating, auto-increment values will be reset

DROP vs TRUNCATE vs DELETE

TRUNCATE

when truncating, auto-increment values will be reset

column_1
1
2
3
4
…
10

DROP vs TRUNCATE vs DELETE

TRUNCATE

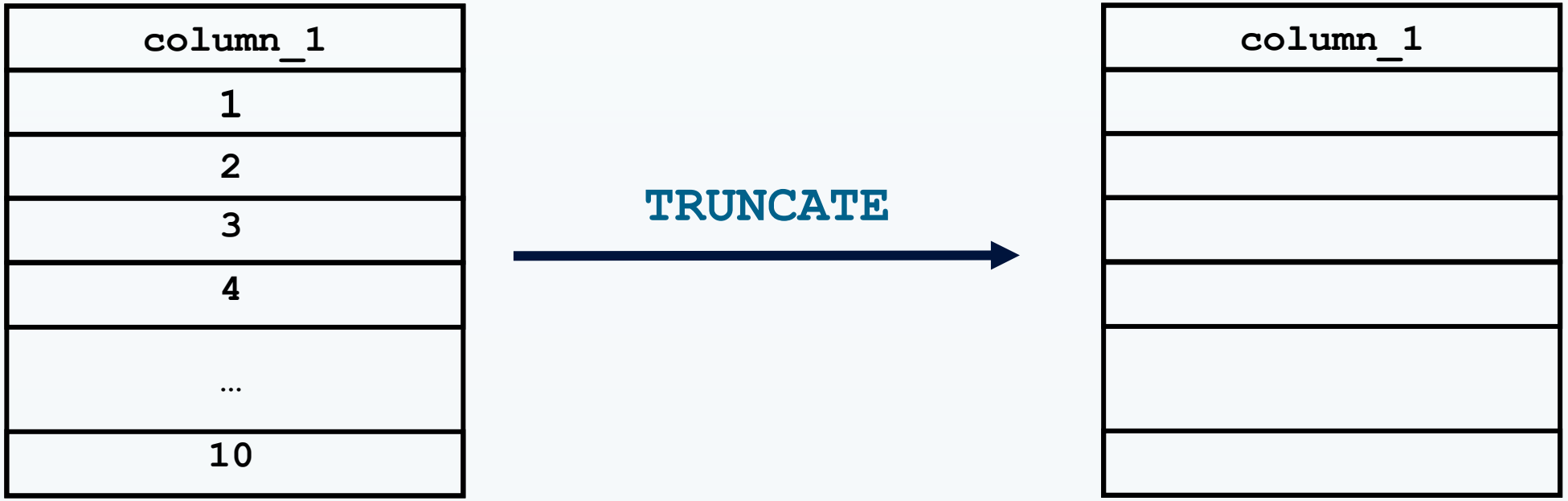
when truncating, auto-increment values will be reset

column_1
1
2
3
4
...
10

DROP vs TRUNCATE vs DELETE

TRUNCATE

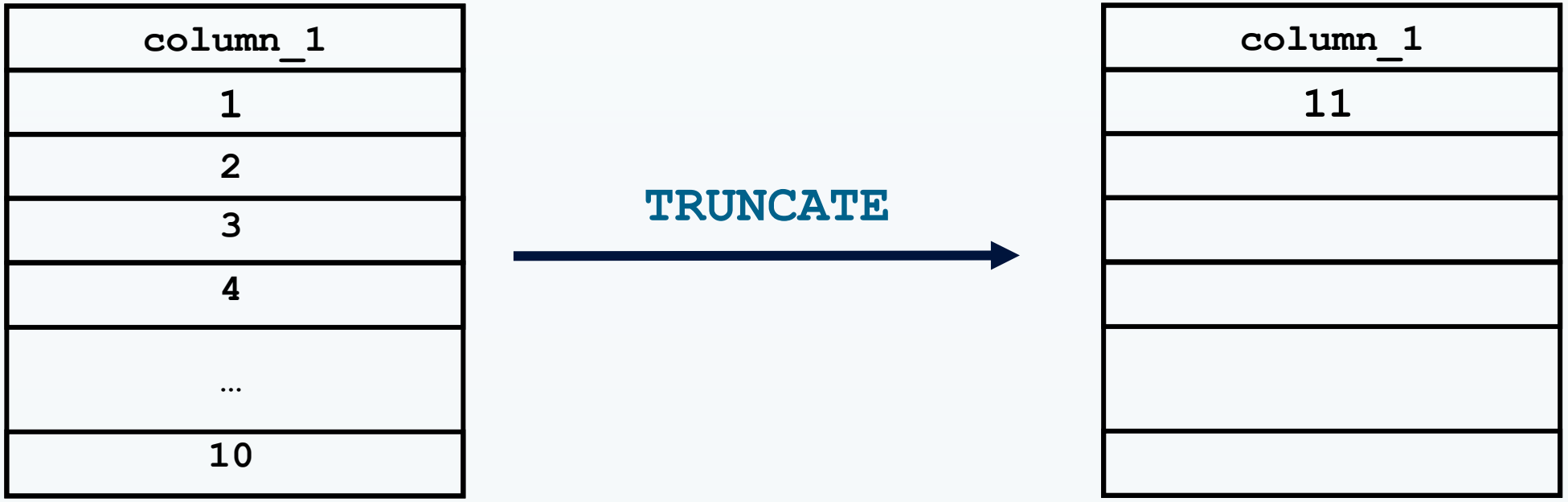
when truncating, auto-increment values will be reset



DROP vs TRUNCATE vs DELETE

TRUNCATE

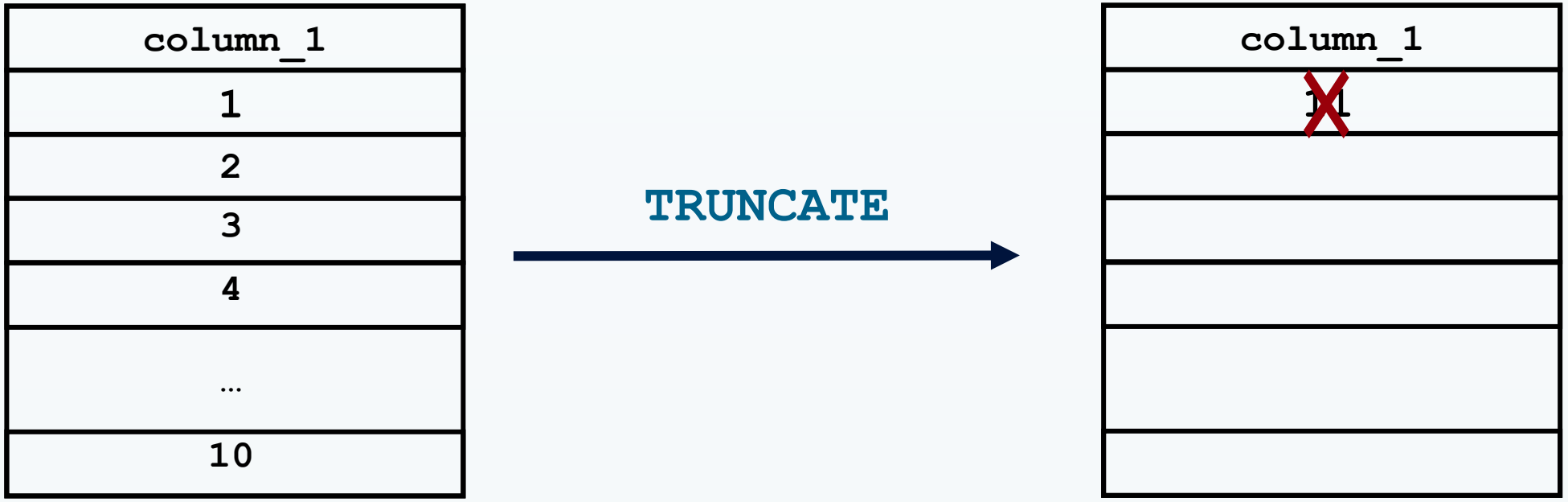
when truncating, auto-increment values will be reset



DROP vs TRUNCATE vs DELETE

TRUNCATE

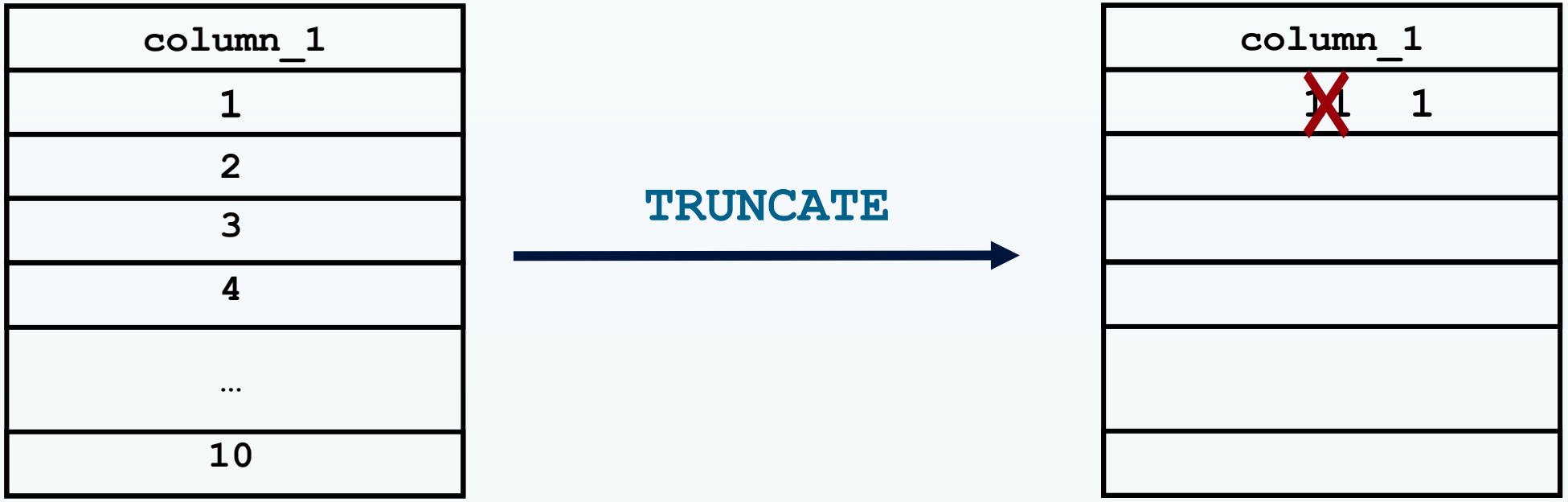
when truncating, auto-increment values will be reset



DROP vs TRUNCATE vs DELETE

TRUNCATE

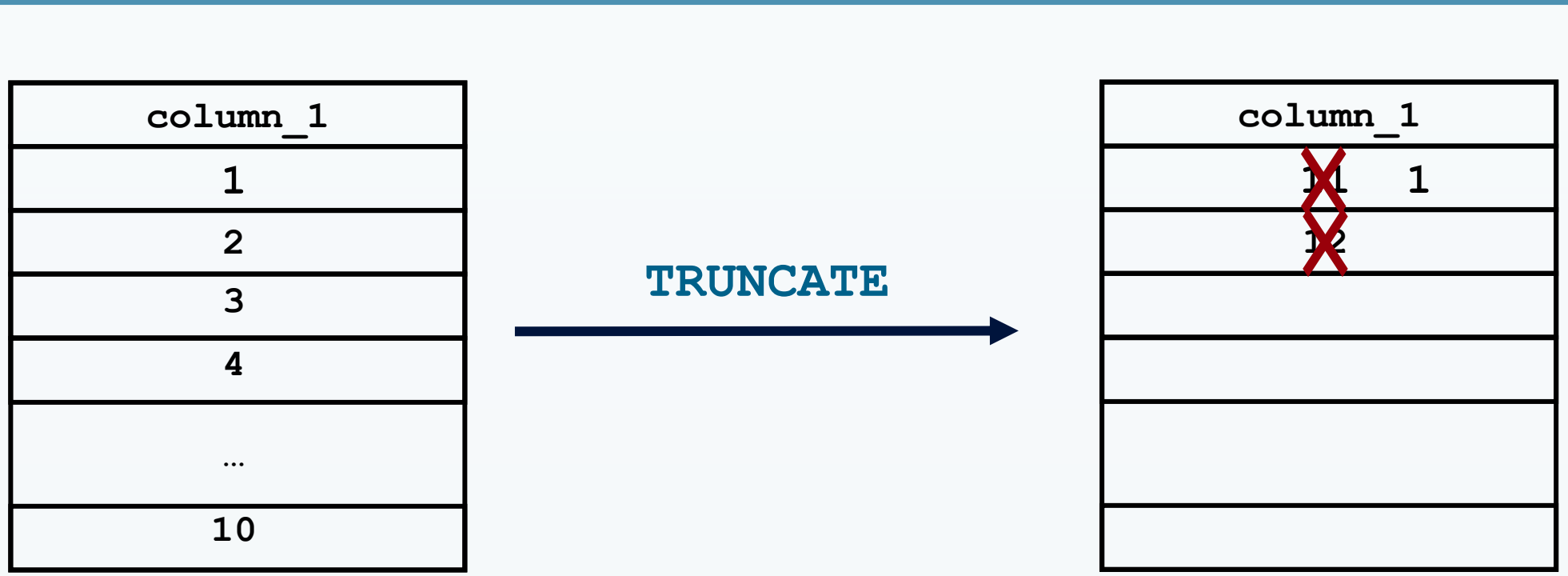
when truncating, auto-increment values will be reset



DROP vs TRUNCATE vs DELETE

TRUNCATE

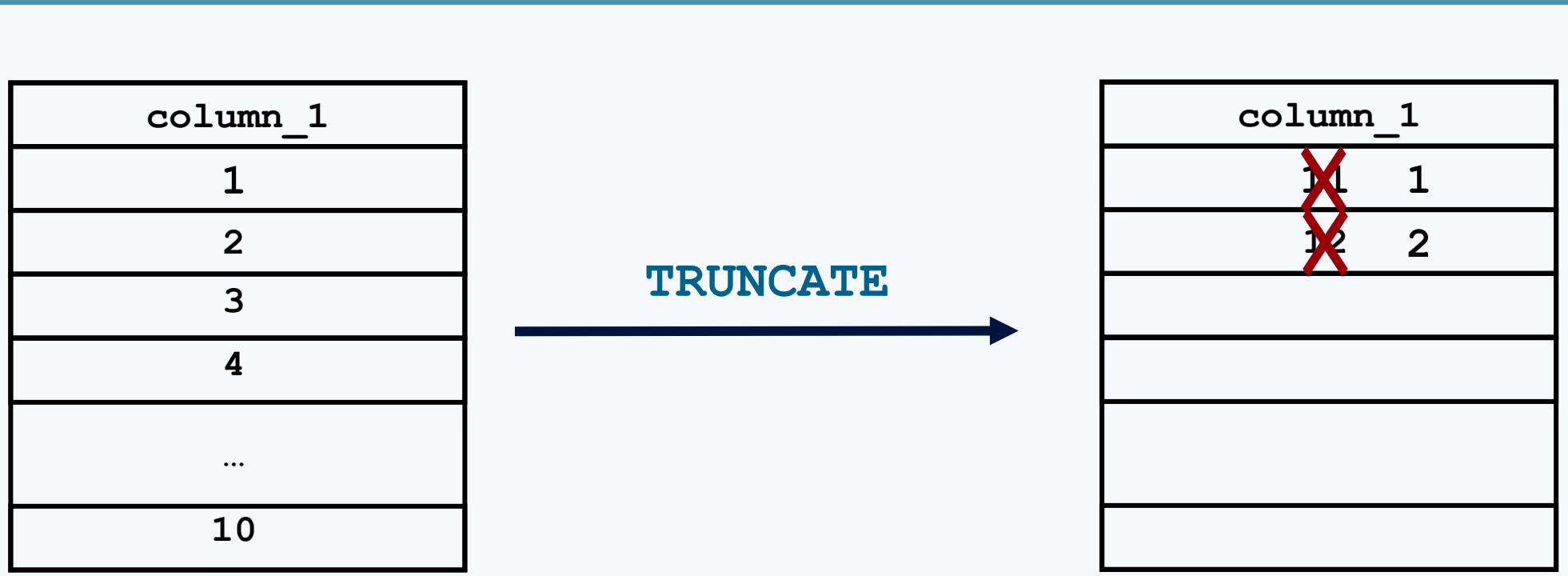
when truncating, auto-increment values will be reset



DROP vs TRUNCATE vs DELETE

TRUNCATE

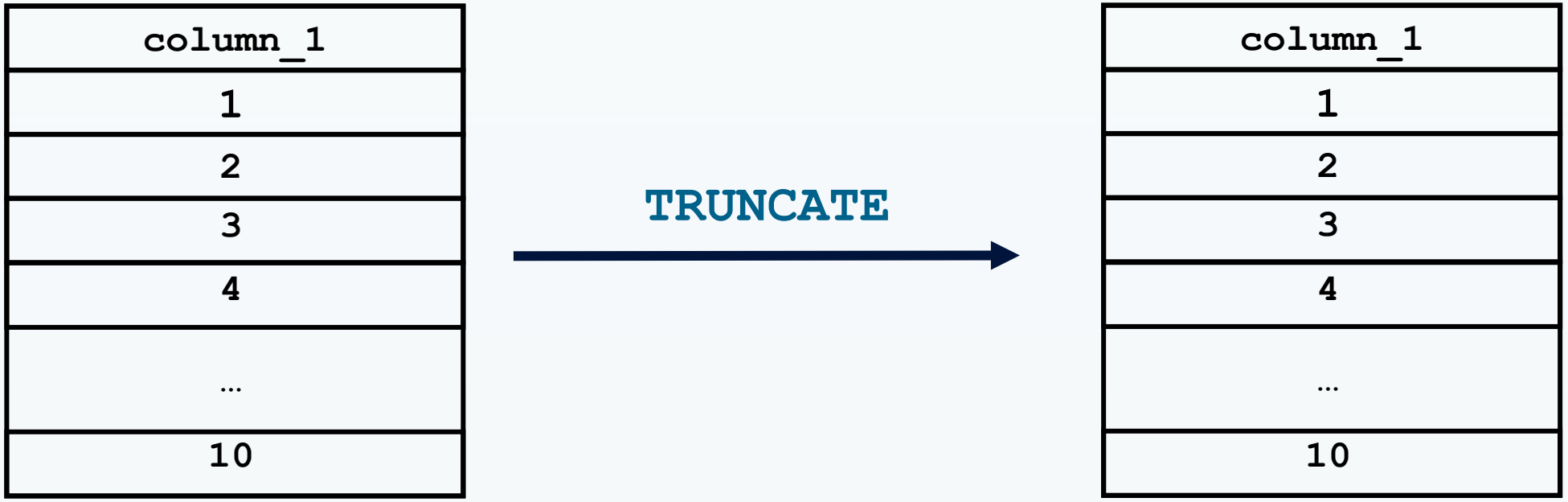
when truncating, auto-increment values will be reset



DROP vs TRUNCATE vs DELETE

TRUNCATE

when truncating, auto-increment values will be reset



DROP vs TRUNCATE vs DELETE

DELETE

removes records *row by row*


```
DELETE FROM table_name
WHERE conditions;
```

TRUNCATE ~ DELETE without WHERE

DROP vs TRUNCATE vs DELETE

TRUNCATE vs DELETE without WHERE

- the SQL optimizer will implement different programmatic approaches when we are using TRUNCATE or DELETE

TRUNCATE delivers the output much *quicker* than DELETE

row by row *row by row*

DROP vs TRUNCATE vs DELETE

TRUNCATE vs DELETE without WHERE

- the SQL optimizer will implement different programmatic approaches when we are using TRUNCATE or DELETE

TRUNCATE delivers the output much *quicker* than DELETE

~~*row by row*~~

row by row

DROP vs TRUNCATE vs DELETE

TRUNCATE vs DELETE without WHERE

- auto-increment values are *not* reset with DELETE

DROP vs TRUNCATE vs DELETE

TRUNCATE vs DELETE without WHERE

- auto-increment values are *not* reset with DELETE

column_1
1
2
3
4
…
10

DROP vs TRUNCATE vs DELETE

TRUNCATE vs DELETE without WHERE

- auto-increment values are *not* reset with DELETE

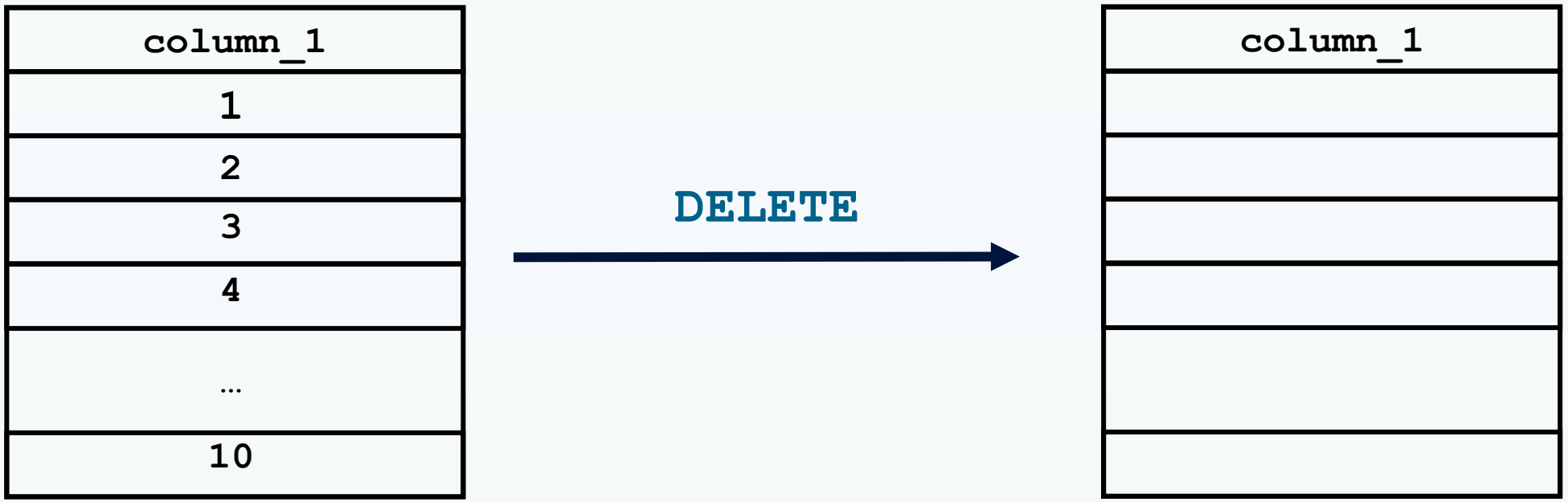
column_1
1
2
3
4
…
10

DELETE

DROP vs TRUNCATE vs DELETE

TRUNCATE vs DELETE without WHERE

- auto-increment values are *not* reset with DELETE



DROP vs TRUNCATE vs DELETE

TRUNCATE vs DELETE without WHERE

- auto-increment values are *not* reset with DELETE