



The SQL DELETE Statement

A modern conference room with large windows and a long table. The room is empty, with several chairs arranged around the table. The view outside the windows shows a cityscape. The image has a blue tint and a stylized, torn-paper-like border.

The DELETE Statement

The DELETE Statement

- the DELETE statement

removes records from a database



SQL

```
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 TRUNCATE vs DELETE

DROP vs TRUNCATE vs DELETE

column_1
1
2
3
4
...
10

DROP vs TRUNCATE vs DELETE

DROP

column_1
1
2
3
4
...
10

DROP vs TRUNCATE vs DELETE

DROP

1
2
3
4
...
10

+

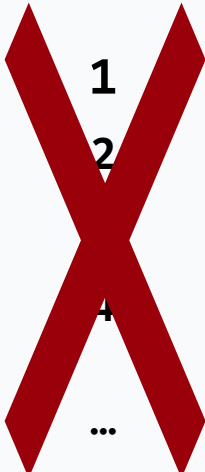
column_1

+

indexes
constraints
...


DROP vs TRUNCATE vs DELETE

DROP

 1
2
4
...
10

+

column_1



+

 indexes
constraints
...

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

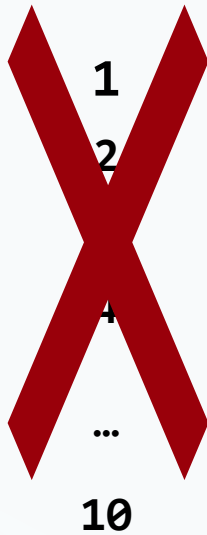
1
2
3
4
...
10

+

column_1

DROP vs TRUNCATE vs DELETE

- TRUNCATE ~ DELETE without WHERE


1
2
4
...
10

+

column_1

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

TRUNCATE



DROP vs TRUNCATE vs DELETE

TRUNCATE

when truncating, auto-increment values will be reset

column_1
1
2
3
4
...
10

TRUNCATE



column_1

DROP vs TRUNCATE vs DELETE

TRUNCATE

when truncating, auto-increment values will be reset

column_1
1
2
3
4
...
10

TRUNCATE



column_1
11

DROP vs TRUNCATE vs DELETE

TRUNCATE

when truncating, auto-increment values will be reset

column_1
1
2
3
4
...
10

TRUNCATE

column_1
1

DROP vs TRUNCATE vs DELETE

TRUNCATE

when truncating, auto-increment values will be reset

column_1
1
2
3
4
...
10

TRUNCATE

column_1
1 1

DROP vs TRUNCATE vs DELETE

TRUNCATE

when truncating, auto-increment values will be reset

column_1
1
2
3
4
...
10

TRUNCATE

column_1
1 1
2

DROP vs TRUNCATE vs DELETE

TRUNCATE

when truncating, auto-increment values will be reset

column_1
1
2
3
4
...
10

TRUNCATE

column_1
1 1
2 2

DROP vs TRUNCATE vs DELETE

TRUNCATE

when truncating, auto-increment values will be reset

column_1
1
2
3
4
...
10

TRUNCATE



column_1
1
2
3
4
...
10

DROP vs TRUNCATE vs DELETE



DELETE

DROP vs TRUNCATE vs DELETE

- DELETE

removes records *row by row*

DROP vs TRUNCATE vs DELETE

- DELETE

removes records *row by row*



SQL

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

DROP vs TRUNCATE vs DELETE

- DELETE

removes records *row by row*



SQL

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

- TRUNCATE ~ DELETE without WHERE

DROP vs TRUNCATE vs DELETE

TRUNCATE vs 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

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



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

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

column_1
1
2
3
4
...
10

DELETE



column_1

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



column_1
11
12
13
14
...
20

Next:

Next:

SQL Functions