SQL DELETE & DROP

DELETE

Syntax

```
DELETE FROM table_name
WHERE some_column = some_value;
```

Example

```
DELETE FROM cars
WHERE year IS NULL;
```

	id [PK] integer	year integer	make text	model text	wheel_count integer
1	1	2020	Toyota	Prius	4
2	2	2012	The Ford Motor Company	Focus	4
3	3	2020	Nissan	Altima	4
4	4	[null]	Elio	P5	3

	id [PK] integer	year integer	make text	model text	wheel_count integer
1	1	2020	Toyota	Prius	4
2	2	2012	The Ford Motor Company	Focus	4
3	3	2020	Nissan	Altima	4

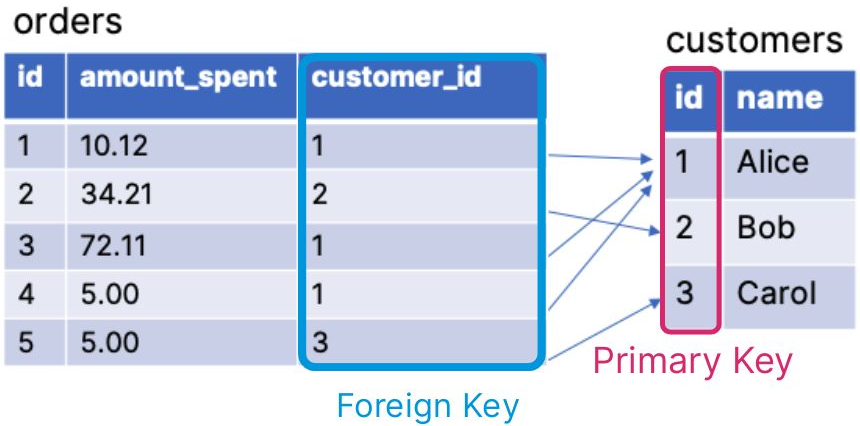
NOTE: Use IS / IS NOT instead of = / != when comparing to null

ON DELETE

Problem

Need to preserve foreign key integrity when removing rows

ON DELETE



Example

One-to-many relationship: customers to orders

Assume keys are non-nullable

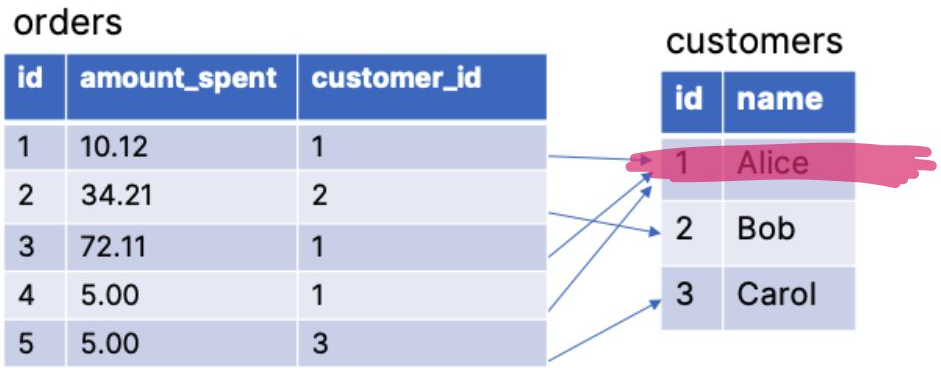
ON DELETE

Deleting an order is trivial

A customer can exist without an order

An order cannot exist without a customer

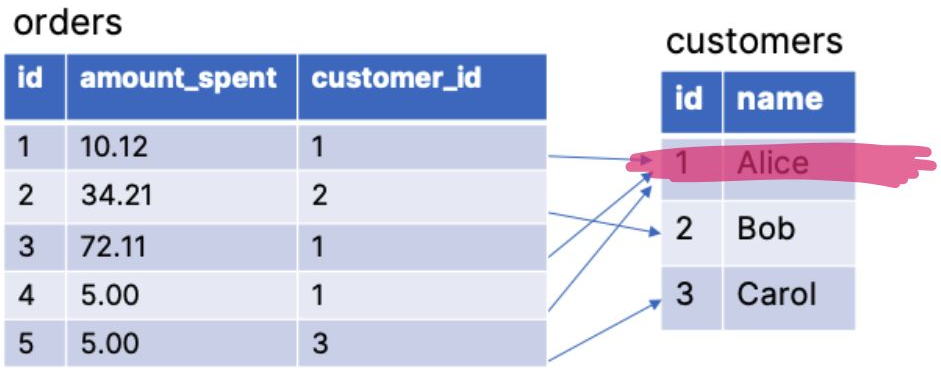
ON DELETE



If we delete Alice, what happens to orders 1, 3, and 4?

Deleting a customer requires handling that customer's orders, to avoid a foreign key constraint violation

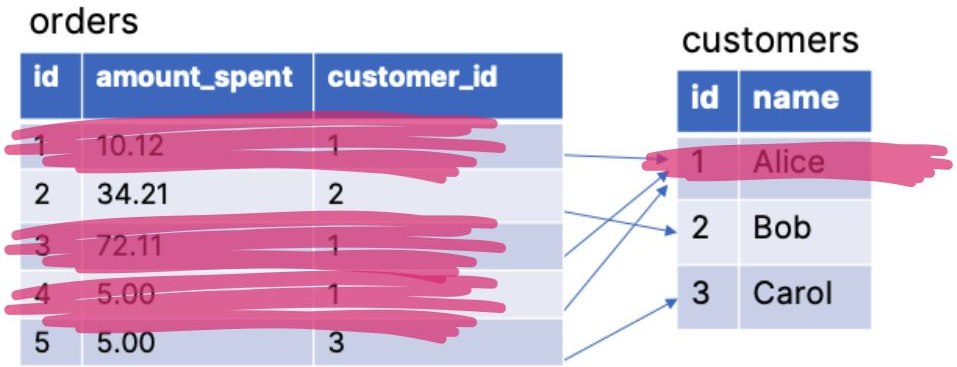
ON DELETE



Problem

Handle customer's orders when deleting customer

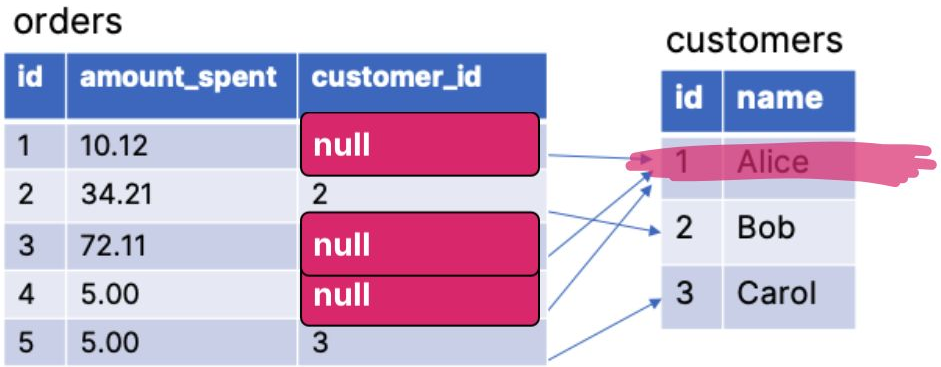
ON DELETE



Solutions

CASCADE: delete records

ON DELETE

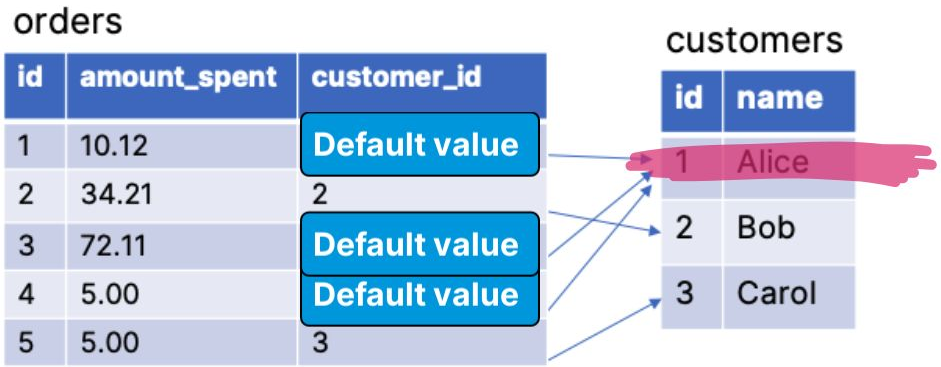


Solutions

CASCADE: delete records

SET NULL: set foreign key to NULL (key must be nullable)

ON DELETE



Solutions

CASCADE: delete records

SET NULL: set foreign key to NULL (key must be nullable)

SET DEFAULT: set foreign key to default value

ON DELETE

orders

id	amount_spent	customer_id
1	10.12	1
2	34.21	2
3	72.11	1
4	5.00	1
5	5.00	3

customers

id	name
1	Alice
2	Bob
3	Carol

Remove customer's orders on delete

```
CREATE TABLE orders(
    id SERIAL PRIMARY KEY,
    amount_spent NUMERIC NOT NULL,
    CONSTRAINT fk_customer
        FOREIGN KEY(customer_id)
        REFERENCES customers(id)
        ON DELETE CASCADE
);
```

Column is non-nullable

```
CREATE TABLE orders(
    id SERIAL PRIMARY KEY,
    amount_spent NUMERIC NOT NULL,
    CONSTRAINT fk_customer
        FOREIGN KEY(customer_id)
        REFERENCES customers(id)
        ON DELETE SET NULL
);
```

Reassigning orders is not sensible

```
CREATE TABLE orders(
    id SERIAL PRIMARY KEY,
    amount_spent NUMERIC NOT NULL,
    CONSTRAINT fk_customer
        FOREIGN KEY(customer_id)
        REFERENCES customers(id)
        ON DELETE SET DEFAULT 1
);
```

orders

id	amount_spent	customer_id
1	10.12	1
2	34.21	2
3	72.11	1
4	5.00	1
5	5.00	3

customers

id	name
1	Alice
2	Bob
3	Carol

Remove customer's orders on delete

```
CREATE TABLE orders(
    id SERIAL PRIMARY KEY,
    amount_spent NUMERIC NOT NULL,
    CONSTRAINT fk_customer
        FOREIGN KEY(customer_id)
        REFERENCES customers(id)
        ON DELETE CASCADE
);
```

Column is non-nullable

```
CREATE TABLE orders(
    id SERIAL PRIMARY KEY,
    amount_spent NUMERIC NOT NULL,
    CONSTRAINT fk_customer
        FOREIGN KEY(customer_id)
        REFERENCES customers(id)
        ON DELETE SET NULL
);
```

Reassigning orders is not sensible

```
CREATE TABLE orders(
    id SERIAL PRIMARY KEY,
    amount_spent NUMERIC NOT NULL,
    CONSTRAINT fk_customer
        FOREIGN KEY(customer_id)
        REFERENCES customers(id)
        ON DELETE SET DEFAULT 1
);
```

TRUNCATE

TRUNCATE will remove all records, but not the table itself

Example

```
TRUNCATE TABLE cars;
```

DROP

DROP table:

```
DROP TABLE cars;
```

DROP database (don't try this)

```
DROP DATABASE week2;
```