

# No brown M&Ms

On foreign keys, isolation levels and Van Halen

Ben Darnell, Chief Architect and Co-Founder



# Foreign key

order			tpcc.order_line	
order_id 🔑	int8 <mark>NN</mark>	1 *	order_id 🔑	int8 <mark>NN</mark>
customer_id	int8		ol_number 🔑	int8 NN
entry_date	timestamp		item_id	int8 NN
status	enum		delivery_date	timestamp
			quantity	int8
			amount	decimal(6,2)

# Foreign key constraint

```
ALTER TABLE order_line
ADD CONSTRAINT fkey_order_line_1
FOREIGN KEY(order_id)
REFERENCES orders(order_id)
```





# INSERT with foreign key constraints

```
INSERT INTO order_line
    VALUES (...);
COMMIT;
```

# Manual foreign key constraints

```
SELECT order_id
   FROM orders
   WHERE order_id = $1;
   --if not found: return

INSERT INTO order_line
   VALUES (...);

COMMIT;
```

### **Application constraints**

```
SELECT order_id
   FROM orders
   WHERE order_id = $1
   AND status = 'open';
   --if not found: return
INSERT INTO order line
   VALUES (...);
COMMIT;
```

#### Concurrency

Safe in SERIALIZABLE isolation

```
SELECT order id
   FROM orders
   WHERE order_id = $1;
   AND status = 'open';
   --if not found: return
INSERT INTO order line
   VALUES (...);
COMMIT;
```

```
UPDATE orders SET status = 'paid'
   WHERE order_id = $1;
COMMIT:
```

### **Explicit locking**

Supports READ COMMITTED and SNAPSHOT isolation

```
SELECT order id
   FROM orders FOR SHARE
   WHERE order_id = $1;
   AND status = 'open';
   --if not found: return
                                 UPDATE orders SET status = 'paid'
INSERT INTO order line
                                    WHERE order_id = $1;
   VALUES (...);
                                    --waits to acquire lock
COMMIT:
                                 COMMIT;
```

