PostgreSQL Pitfalls

Jeff Davis, Truviso, Inc.
What's a “Pitfall”?  

- Unintuitive  
- Frequently misused or misunderstood  
- Not necessarily “wrong”  
- Might even be required by SQL standard
• SELECT * FROM foo WHERE x NOT IN (SELECT y FROM bar);  
• Works great during testing.
NOT IN (...) Pitfall

• SELECT * FROM foo WHERE x NOT IN (SELECT y FROM bar);
• Works great during testing.
• As soon as you put a NULL into “bar”, query never returns any results at all!
  • IN uses SQL 3-valued logic
  • Predicate evaluates to FALSE or NULL
Interval Math

- Add/subtract interval from timestamp
  - SELECT * FROM order WHERE ts > (now() - '1 day'::interval);
- Multiply interval by a scalar
  - SELECT x * '1 second'::interval + now() FROM foo;
- Wow, that's easy!
Additive inverse doesn't work:

```sql
SELECT
  '2009-03-31'::timestamp
- '1 month '::interval
+ '1 month '::interval
<> '2009-03-31'::timestamp
```
Interval Math Pitfall 2

• Commutativity is lost:

```sql
select '2009-02-28'::timestamp + '1 month'::interval + '1 day'::interval <> '2009-02-28'::timestamp + '1 day'::interval + '1 month'::interval;
```
• Scalar multiplication is not the same as repeated addition:

```sql
select
  '2009-01-31'::timestamp + '2 month'::interval
<>
  '2009-01-31'::timestamp + '1 month'::interval + '1 month'::interval;
```
CREATE TABLE t(i INT UNIQUE);
INSERT INTO t VALUES(1);
INSERT INTO t VALUES(2);
INSERT INTO t VALUES(3);
UPDATE t SET i = i + 1;

• ERROR!
  • Finds duplicate key and raises error before command is finished.
UNIQUE Pitfall (cont.)

UPDATE t SET i = i + 1;

• ERROR!
  Finds duplicate key and raises error before command is finished.

Update

Conflict!
CREATE TABLE foo(b BOOLEAN);
INSERT INTO foo VALUES(TRUE);
INSERT INTO foo VALUES(FALSE);
Snapshots Pitfall

• Transaction 1:
  BEGIN;
  UPDATE foo SET b = NOT b;
  COMMIT;

• Transaction 2:
  SELECT * FROM foo WHERE b FOR SHARE;

  Returns Nothing!
Snapshots Pitfall (cont.)

Update

T2 sees (waiting for T1)
Snapshots Pitfall (cont.)

Update

T2 sees (waiting for T1)

WHERE b
After T1 commit, follow update chain.

T2 sees

WHERE b
Snapshots Pitfall (cont.)

TRUE
FALSE
FALSE
TRUE

T2 Sees
WHERE b
Good News!

Deferrable UNIQUE check patch submitted by Dean Rasheed

• July Commitfest
• commitfest.postgresql.org
Conclusion

• PostgreSQL isn't perfect
• SQL isn't perfect
• Even if they were, these complex behaviors are tricky
  • E.g. Interval math
• Where improvements can be made, solutions are appearing very quickly