CONSTRAINTS

A DEVELOPER'S SECRET WEAPON

PG Day Paris 2018-03-15

WILL LEINWEBER
@LEINWEBER
CITUSDATA.COM

INTRO

CONSTRAINTS

maybe not the most exciting topic

just want DB to safely store&retrieve data

stern parent saying "No!"

RAILS

changed the landscape

before: spaghetti or mountains of xml

after: convention instead of configuration

RAILS

embrace constraints

separate code concerns

pluralize table names

primary key is called "id"

…etc

RAILS

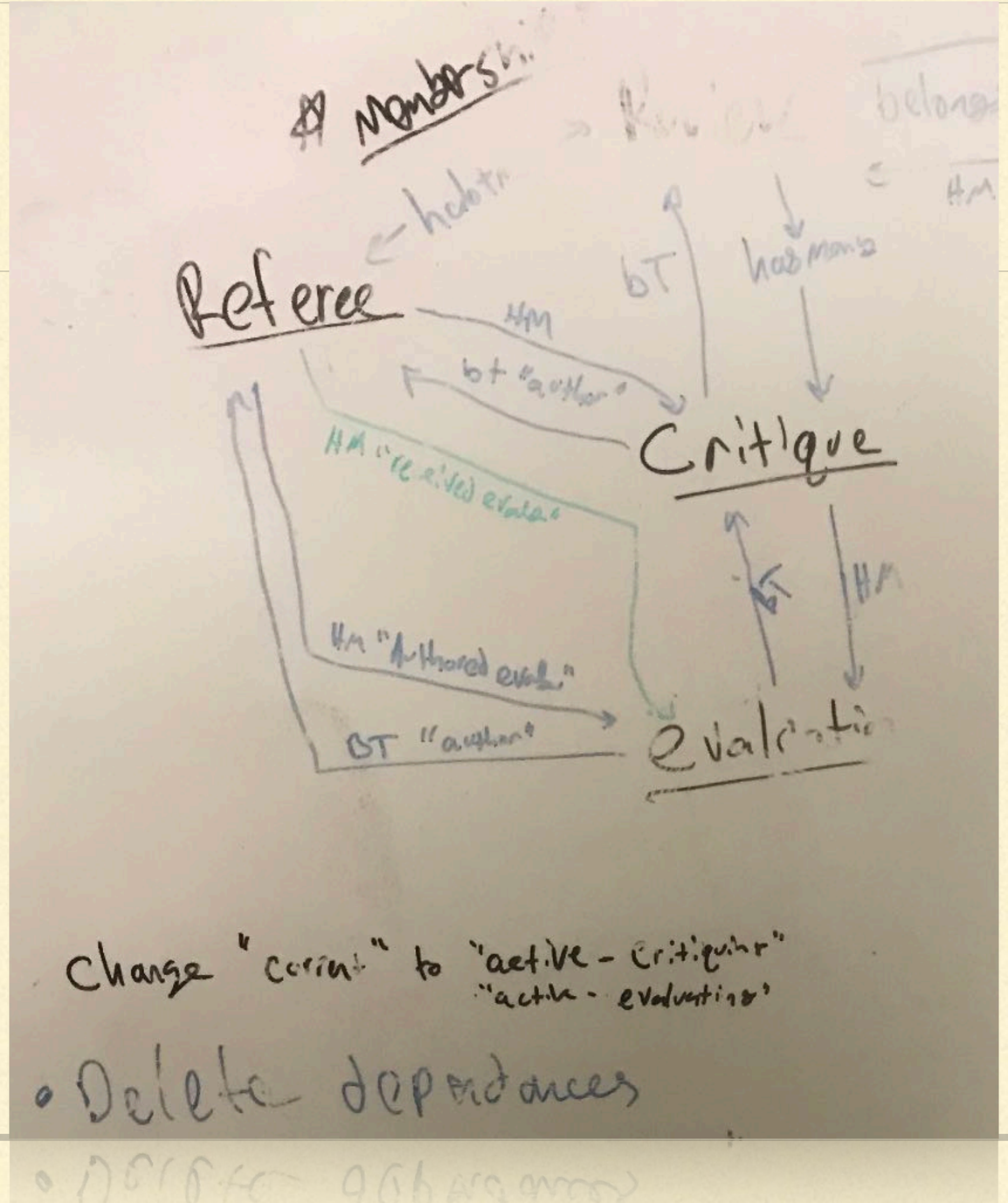
"big dumb hash in the sky"

MEANWHILE...

learning how to make web applications

LEARNING 3NF

examples hard to extrapolate to my problem


Members
Referee
HM
bt "author"
bT
Critique
HM "active evaluator"
HM "Authored eval"
bT "author"
Change "current" to "active-critiquer"
"active-evaluation"
• Delete dependances

TODAY

think about database schema first

work backwards to models and api

MOTIVATION

database is the last line of defense

MOTIVATION

code change frequency >>> schema change frequency

MOTIVATION

~1 year old app	after ~2 years
71 migrations	90 migrations
1203 releases	1454 releases

MOTIVATION

logical corruption is more likely to come from app bug

BUGS

bugs that can be caught by schema are particularly dangerous

BUGS

delayed problem

hard to find cause

can last for months

CLEANUP

MOTIVATION

don't write bad data in the first place

TRADITIONAL CONSTRAINTS

TRADITIONAL CONSTRAINTS

```
NOT NULL
```

TRADITIONAL CONSTRAINTS

```
CREATE UNIQUE INDEX
ON users (email);
```

PARTIAL UNIQUE

```
CREATE UNIQUE INDEX
  ON users (email)
  WHERE deleted_at IS NULL;
```

ENFORCE ASSUMPTIONS

"This property *should* always be a fibonacci number"

"*Should* never be…"

DATATYPES

DATATYPES

not often thought of as constraints

constrain what type of data gets in

DATATYPES

numbers are actually numbers

booleans are actually booleans

…etc

DATATYPES

…other databases

ENUMS

```
CREATE TYPE state
AS ENUM ('creating', 'running');


ALTER TYPE state
ADD VALUE 'deleting';
```

RANGES

```
SELECT i, i <@ '[1,10)'::int4range included
FROM (VALUES (1),(5),(10)) as v(i);

 i  | included
----+----------
  1 | t
  5 | t
 10 | f
```

RANGES

int4range, int8range, numrange

~~tsrange~~, tstzrange

daterange

RANGES

SMALL OPTIMIZATION

```
CREATE TABLE somename (
  aws_id text COLLATE "C" NOT NULL
);
-- i-0598c7d356eba48d7
```

OTHER TYPES

uuid

macaddr, inet, cidr

array, hstore

geometric

DATATYPES

downside of using JSONB

FOREIGN KEYS

FOREIGN KEYS

```
CREATE TABLE posts(
  user_id int NOT NULL
    REFERENCES users(id),
  …
```

OPTIONS

```
REFERENCES foo ON DELETE/UPDATE

    NO ACTION  / RESTRICT
    CASCADE
    SET NULL / DEFAULT
```

PROBLEMS WITH TESTING

transaction vs. truncation/deleting

```
SET CONSTRAINTS all DEFERRED;
```

CHECK

CHECK

custom logic

ENSURE POSITIVE NUMBER

```
CREATE TABLE products(
   name  text,
   price int CHECK(price > 0)
);
```

REFERENCE OTHER COLUMN

```
CREATE TABLE products(
   name    text,
   price   int CHECK(price > 0),
   sale_p int CHECK(sale_p > 0),
   CHECK (price > sale_p)
);
```

PERCENTS

```
scale float DEFAULT 1.0 NOT NULL,
CHECK (scale >= 0 && scale <= 1)
```

MORE

```
CHECK ((json_col->>'i_prop')::int > 0)
```

USER DEFINED FUNCTIONS

```
CREATE OR REPLACE FUNCTION is_fib(i int) RETURNS boolean AS $$
DECLARE
  a integer := 5*i*i+4;
  b integer := 5*i*i-4;
  asq integer := sqrt(a)::int;
  bsq integer := sqrt(b)::int;
BEGIN
  RETURN asq*asq=a OR bsq*bsq=b;
end
$$ LANGUAGE plpgsql IMMUTABLE STRICT;
```

USER DEFINED FUNCTIONS

```
# CREATE TABLE onlyfib( i int CHECK (is_fib(i)) );
CREATE TABLE

# insert into onlyfib values (5), (8);
INSERT 0 2

# insert into onlyfib values (6);
ERROR:  new row for relation "onlyfib" violates
check constraint "onlyfib_i_check"
DETAIL:  Failing row contains (6).
```

DOMAINS

```
# CREATE DOMAIN fib AS int CHECK (is_fib(VALUE));
# CREATE TABLE onlyfib(i fib);

# insert into onlyfib values (5),(8);
INSERT 0 2

# insert into onlyfib values (6);
ERROR:  value for domain fib violates check
constraint "fib_check"
```


NEWSSTAND
FIBONACCI PIGEONS

EXCLUSION

&& OVERLAP

```
select '[ 1,10)'::int4range
    && '[15,20)'::int4range;
f

select '[1,10)'::int4range
    && '[9,20)'::int4range;
t
```

EXCLUSION

```
CREATE TABLE billings (
    formation_id    uuid        NOT NULL,
    validity_period tstzrange NOT NULL,
    price_per_month integer     NOT NULL
);
```

EXCLUSION

```
ALTER TABLE billings
ADD CONSTRAINT billings_excl
EXCLUDE USING gist (
  formation_id WITH =,
  validity_period WITH &&
);
```

ERROR MESSAGE

```
ERROR:  conflicting key value
violates exclusion constraint
"constraint name"

DETAIL:  Key (id, range)=(<new
row>) conflicts with existing key
(id, range)=(<existing row>).
```

RECAP

3 TAKEAWAYS

database is your last line of defense

postgres has some really great constraints

datatypes are constraints too

WILL LEINWEBER
@LEINWEBER
CITUSDATA.COM

thank you