SQUIRREL: Testing Database Management Systems with Language Validity and Coverage Feedback

Rui Zhong*, Yongheng Chen+, Hong Hu*+, Hangfan Zhang*, Wenke Lee+ and Dinghao Wu*

*Penn State University , +GeorgiaTech

Why Database Management Systems

- **Popularity**. E.g., there are likely over **one trillion** (1e12) SQLite databases in active use nowadays.

Why Database Management Systems

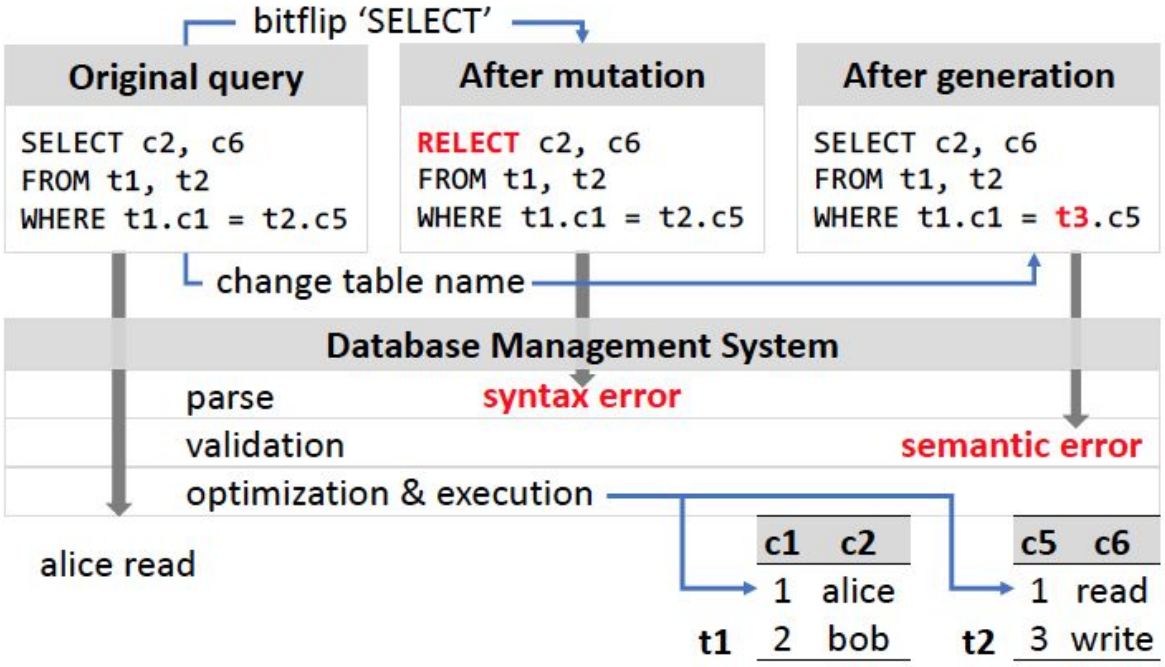
- **Complexity**. E.g., MySQL has over **4 million** LoC. Larger codebases tend to have more bugs.

Challenges in Testing DBMSs

Query processing:

1. Parse
2. Validation
3. Optimization
4. Execution

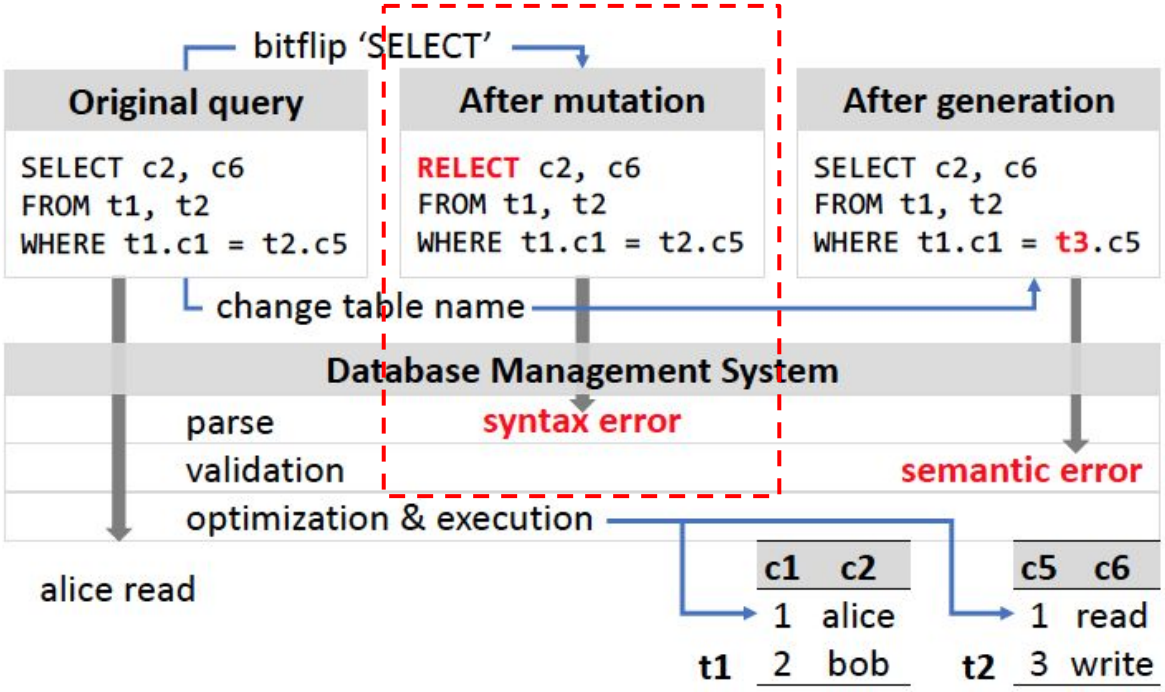
DBMSs check the input queries for syntactic and semantic correctness!



Limitations of Existing Approaches

Mutation-based Fuzzing:

1. Con: syntax-unawared.
2. Pro: adopt feedback mechanism to avoid duplicate efforts.

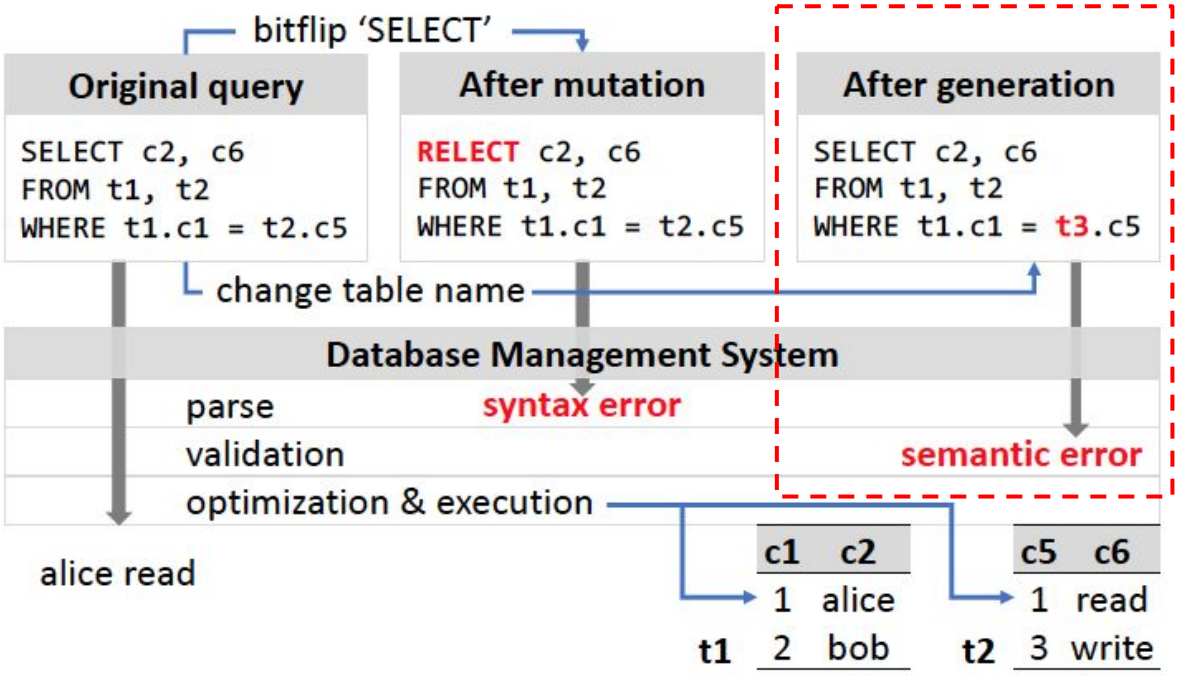


Limitations of Existing Approaches

Generation-based Fuzzing:

1. Pro: syntax-awared.
2. Con: inefficient.

Unable to guarantee **semantic correctness**! :(



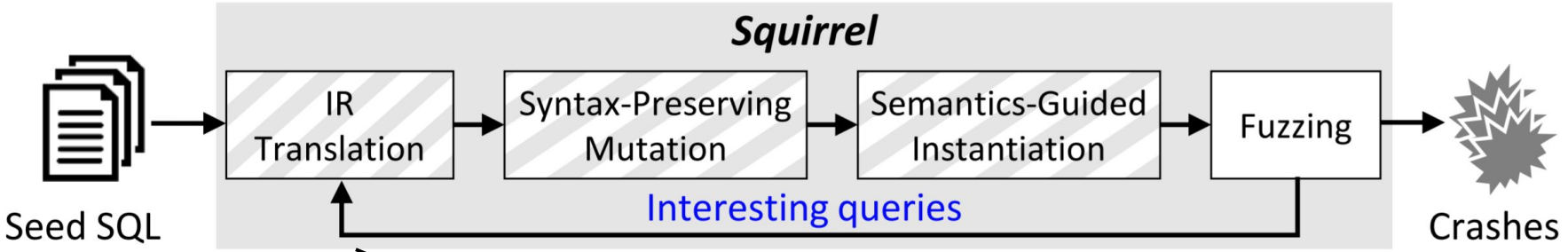
Our Approach: Squirrel

We take advantages of mutation-based and generation-based techniques.

- generate **syntax-correct** queries.
- adopt **feedback mechanism** to prioritize interesting queries.

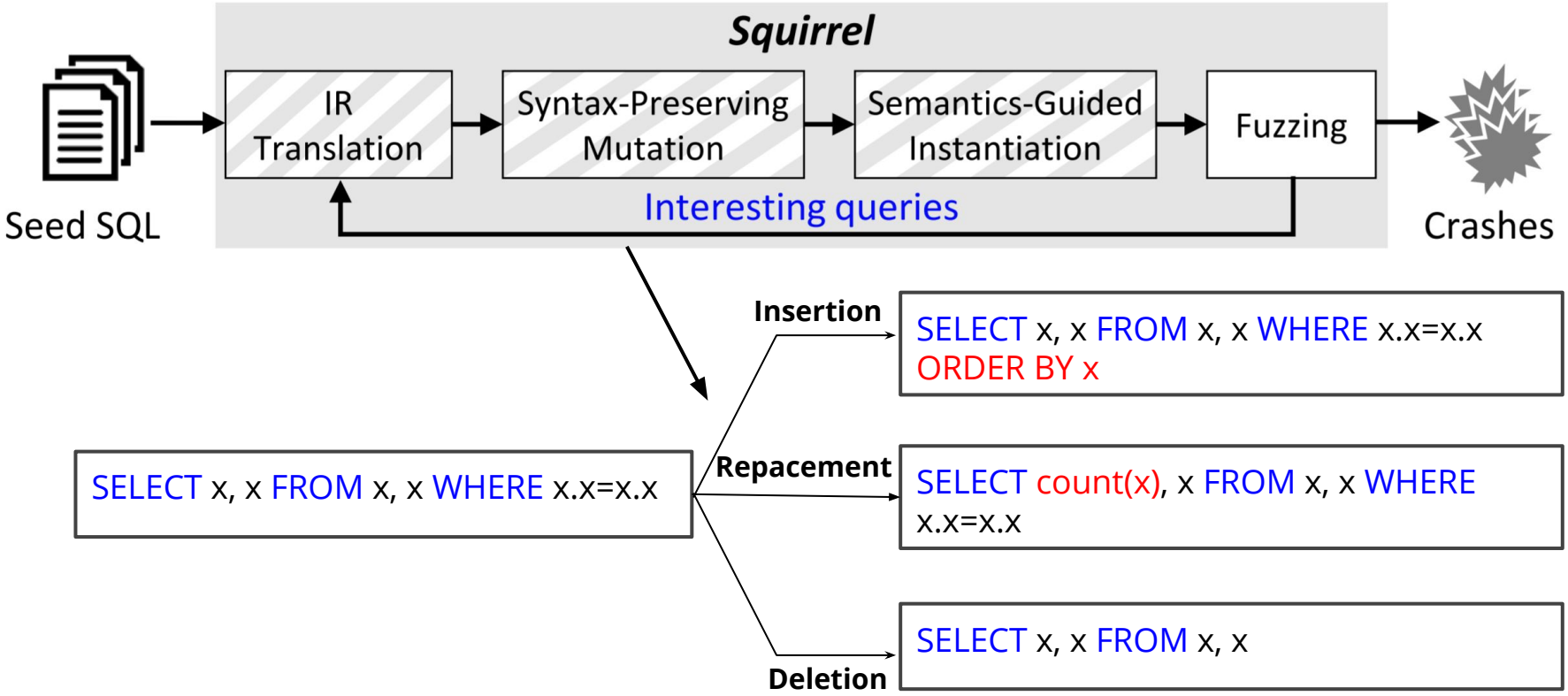
Further, we improve **semantic correctness** to help fuzzer reach deep logics.

Overview of Squirrel

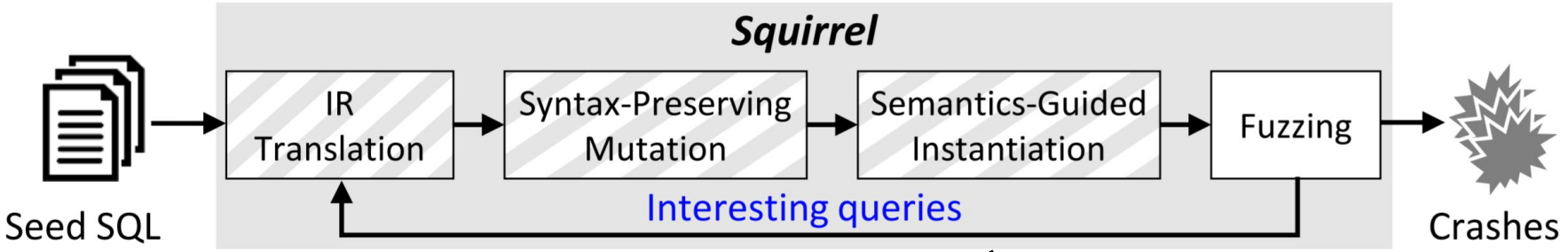



```
v0 = (cname, l=0, r=0, op=0, d="id", t=ColumnName);
v1 = (type, l=0, r=0, op.prefix="INT", d=0);
v2 = (column, l=v0, r=v1, op=0);
v3 = (cname, l=0, r=0, op=0, d="class", t=ColumnName);
v4 = (type, l=0, r=0, op.prefix="INT", d=0);
v5 = (column, l=v3, r=v4, op=0);
v6 = (columns, l=v2, r=v3, op.prefix="(", op.middle=",", op.suffix=")");
…
```

Overview of Squirrel



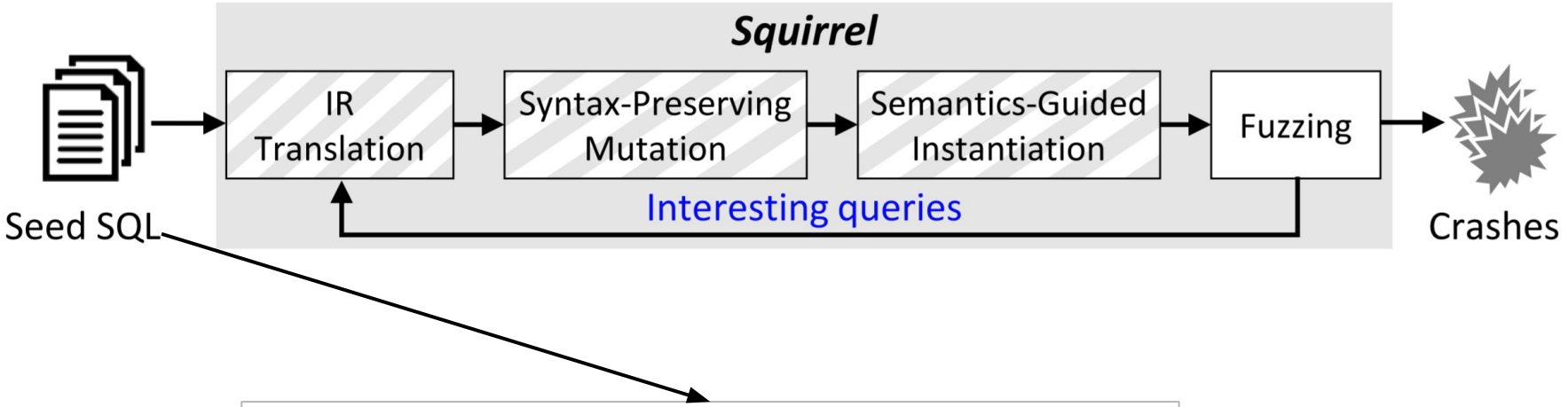
Overview of Squirrel




```
CREATE TABLE x1(x2 INT, x3 INT)
CREATE TABLE x4(x5 INT, x6 INT)
SELECT x7.x8, x9.x10 FROM x11, x12
                    WHERE x13.x14=x15.x16
                    ORDER BY x17
```

```
CREATE TABLE t1(c1 INT, c2 INT)
CREATE TABLE t2(c3 INT, c4 INT)
SELECT t1.c1, t2.c4 FROM t1, t2
                    WHERE t1.c1=t2.c3
                    ORDER BY t1.c1
```

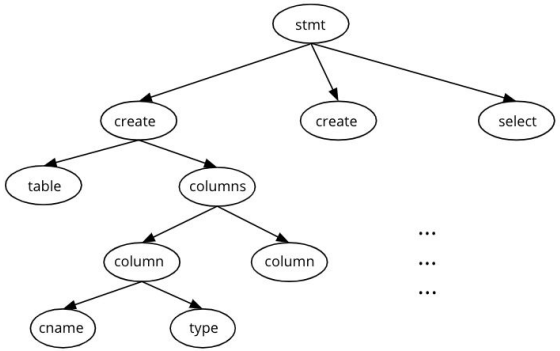
Overview of Squirrel




```
CREATE TABLE stu(id INT, class INT)
CREATE TABLE teacher(id INT, class INT)
SELECT stu.id, teacher.id FROM stu, teacher
                    WHERE teacher.class=stu.class
```

IR Translation

```
CREATE TABLE stu(id INT, class INT)
CREATE TABLE teacher(id INT, class INT)
SELECT stu.id, teacher.id FROM stu, teacher
                    WHERE teacher.class=stu.class
```




```
v0 = (cname, l=0, r=0, op=0, d="id", t=ColumnName);
v1 = (type, l=0, r=0, op.prefix="INT", d=0);
v2 = (column, l=v0, r=v1, op=0);
v3 = (cname, l=0, r=0, op=0, d="class", t=ColumnName);
v4 = (type, l=0, r=0, op.prefix="INT", d=0);
v5 = (column, l=v3, r=v4, op=0);
v6 = (columns, l=v2, r=v3, op.prefix="(", op.middle=",", op.suffix=")");
…
...
```

Mutation: Structure-Data Separation

```
CREATE TABLE stu(id INT, class INT)
CREATE TABLE teacher(id INT, class INT)
SELECT stu.id, teacher.id FROM stu, teacher
                    WHERE teacher.class=stu.class
```

```
CREATE TABLE x1(x2 INT, x3 INT)
CREATE TABLE x4(x5 INT, x6 INT)
SELECT x7.x8, x9.x10 FROM x11, x12
                    WHERE x13.x14=x15.x16
```

Mutation: Insertion, Replacement and Deletion

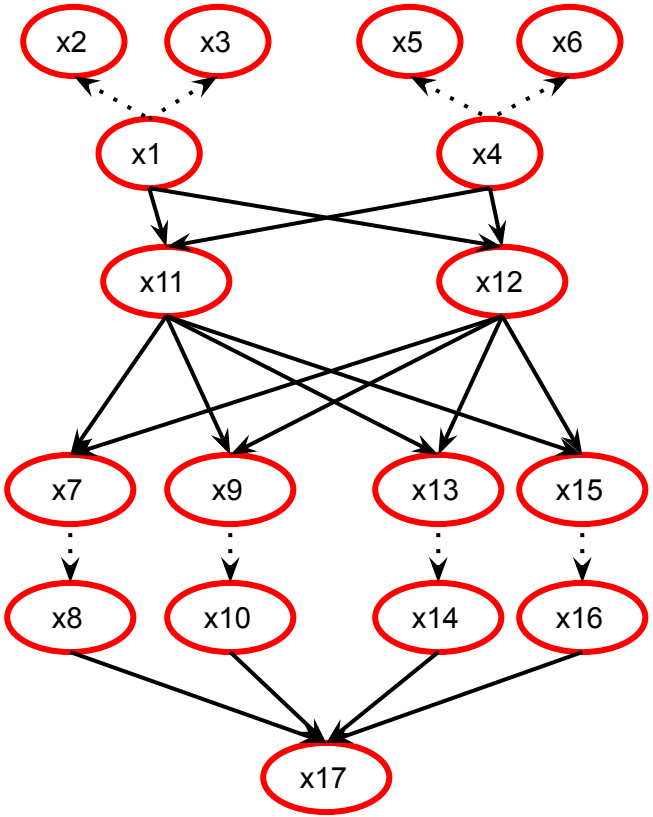
```
CREATE TABLE x1(x2 INT, x3 INT)
CREATE TABLE x4(x5 INT, x6 INT)
SELECT x7.x8, x9.x10 FROM x11, x12
                    WHERE x13.x14=x15.x16
```

```
CREATE TABLE x1(x2 INT, x3 INT)
CREATE TABLE x4(x5 INT, x6 INT)
SELECT x7.x8, x9.x10 FROM x11, x12
                    WHERE x13.x14=x15.x16
                    ORDER BY x17
```

Instantiation

```
CREATE TABLE x1(x2 INT, x3 INT)
CREATE TABLE x4(x5 INT, x6 INT)
SELECT x7, x8, x9, x10 FROM x11, x12
                WHERE x13.x14=x15.x16
                ORDER BY x17
```

```
CREATE TABLE t1(c1 INT, c2 INT)
CREATE TABLE t2(c3 INT, c4 INT)
SELECT t1.c1, t2.c4 FROM t1, t2
                WHERE t1.c1=t2.c3
                ORDER BY t1.c1
```



Evaluation: New Bugs

Ran Squirrel for 40 days on one 16-core server.

Bugs found in SQLite, MySQL and MariaDB

- **63** unique bugs found & confirmed
- **52** bugs fixed
- **12** CVEs assigned

Evaluation: Contributions of Different Aspects

Compared with Squirrel w/o semantic, Squirrel w/o feedback, Squirrel w/o semantic_syntax.

- Feedback helps achieve **2.0x** more new edges.
- Syntax correctness helps achieve up to **1.5x** more new edges.
- Semantic correctness helps achieve up to **1.7x** more new edges.

Evaluation: Compared With Existing Tools

Compared with SQLSmith, Angora, GRIMOIRE, QSYM, AFL.

- up to **10.9x** more edges.
- up to **20.9x** higher syntax correctness.
- up to **243.9x** higher semantic correctness.

Summary

- Squirrel is a general Database Management System testing framework
  - Reach source code at https://github.com/s3team/Squirrel.
- Generate high-quality SQL test cases.
  - well-structured
  - semantic correct
  - efficient
- Discovered bugs in popular DBMSs
  - 63 bugs confirmed
  - 12 CVEs assigned

Q&A