

# Pruebas SQL, NoSQL y Multidispositivo

Javier Tuya

Universidad de Oviedo

Grupo de Investigación en Ingeniería del Software (GIIS)

<http://giis.uniovi.es/>



Alianza Española de Innovación en Software Testing  
San Sebastián, 15 de Diciembre de 2015



# Qué hacemos

## GIIS (Grupo de Investigación en Ingeniería del Software)

- Pruebas de Aplicaciones con Bases de datos
  - Evaluación, generación, reducción
- Pruebas con tecnologías usadas en Big Data Engineering
  - Almacenamiento (NoSQL)
  - Procesamiento (MapReduce)
- Estándares (ISO/IEC/IEEE 29119)
- Formación

Acknowledgements:

**Test4SOA** - *Techniques for Functional Testing in Service Oriented Architectures* (TIN2007-67843-C06-01)

**Test4DBS** - *Test methods for applications with databases and services* (TIN2010-20057-C03-01)

Acknowledgements:

**PERTEST** - *Testing of data persistence and user perspective for new paradigms* (TIN2013-46928-C3-1-R)

# Puedes conocer la cobertura de las pruebas de tu aplicación

Un fragmento simple de un programa Java...

El salario es calculado como un salario base más el promedio de los bonus generados por los proyectos en los que el empleado ha participado

```
public float CalculateSalary(String idEmployee) {
    ResultSet rsTrans;
    Statement stmt;
    String query;
    float basesalary;
    float averagebonus;

    stmt = conn.createStatement();
    query = "SELECT basesalary, avg(bonus) " +
        "FROM Employee E INNER JOIN Assignment A ON E.idEmp = A.idEmp " +
        "INNER JOIN Project P ON A.idProj = P.idProj " +
        "WHERE E.idEmp = " + idEmployee + " AND P.bonus > 0 " +
        "GROUP BY E.idEmp";
    rsTrans = stmt.executeQuery(query);
    rsTrans.next();
    basesalary = rsTrans.getFloat(1);
    averagebonus = rsTrans.getFloat(2);

    return basesalary + averagebonus;
}
```

Ejecutando esta función se cubren  
todas las ramas  
**100% de cobertura?**

# ... y la cobertura de tus SQL?

```
query = "SELECT baselary, avg(bonus) " +  
"FROM Employee E INNER JOIN Assignment A ON E.idEmp = A.idEmp " +  
"INNER JOIN Project P ON A.idProj = P.idProj " +  
"WHERE E.idEmp = " + idEmployee + " AND P.bonus > 0 " +  
"GROUP BY E.idEmp";
```

Qué pasa si...


... Todos los proyectos asignados a un empleado dado tienen  $\text{bonus} > 0$ ?

 El promedio de bonus es correcto

... Algún proyecto asignado tiene  $\text{bonus} = 0$ ?

 El promedio de bonus NO es correcto

... Todos los proyectos asignados tienen  $\text{bonus} = 0$ ?

 La sentencia SQL NO devuelve filas:  
RuntimeException

... El empleado no está asignado a ningún proyecto?

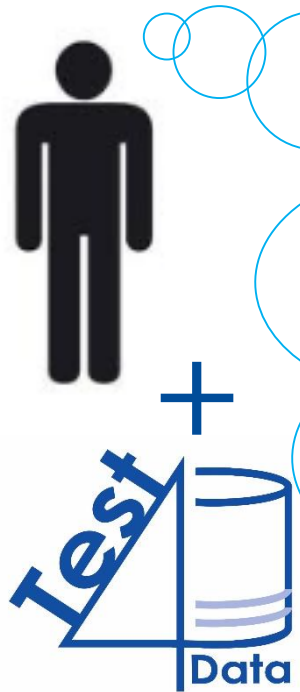
# ... and your SQL test coverage?

```
query = "SELECT basesalary, avg(bonus) " +  
"FROM Employee E INNER JOIN Assignment A ON E.idEmp = A.idEmp " +  
"INNER JOIN Project P ON A.idProj = P.idProj " +  
"WHERE E.idEmp = " + idEmployee + " AND P.bonus > 0 " +  
"GROUP BY E.idEmp";
```

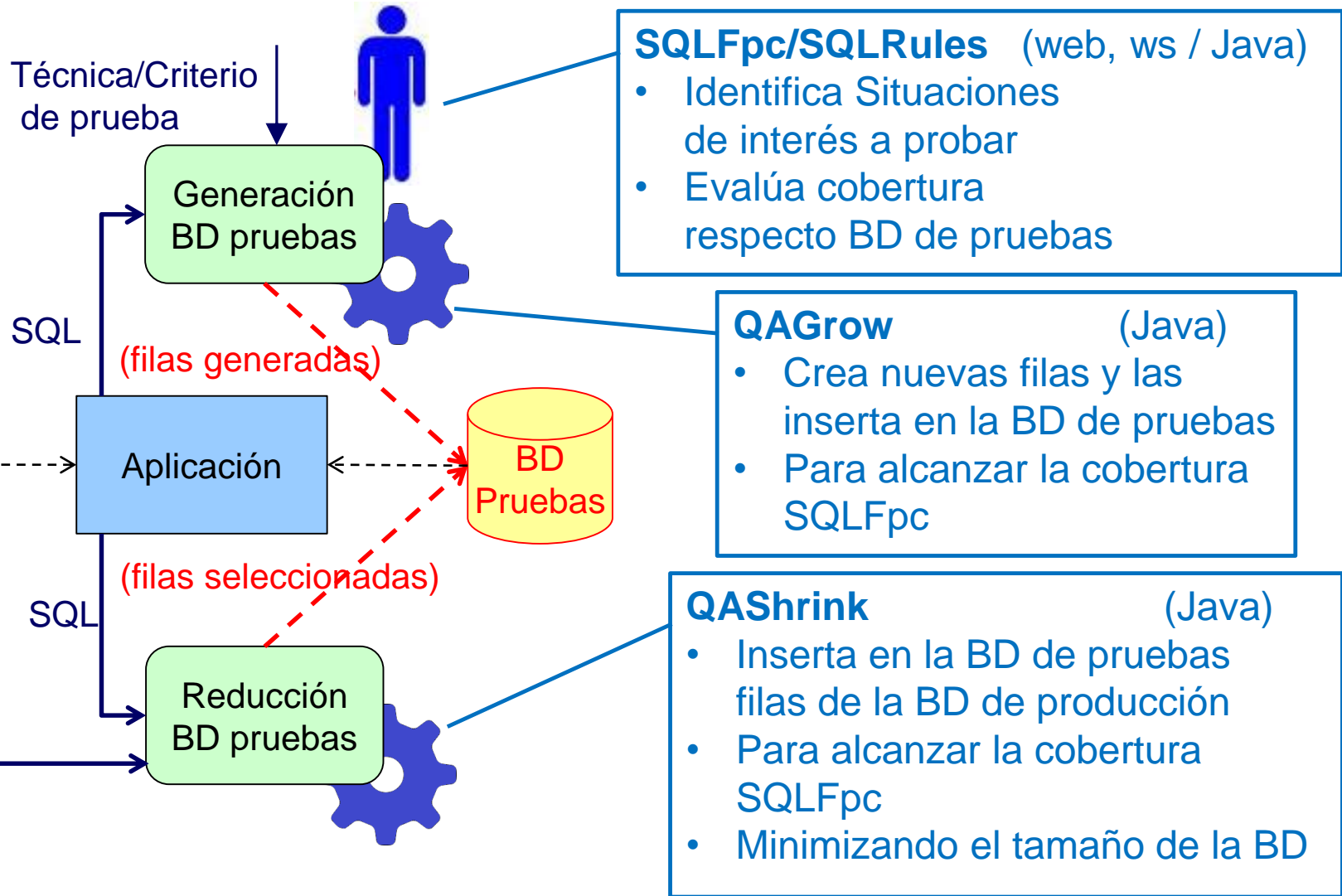
Cómo plantearíamos la prueba?

**Identificar** las diferentes situaciones que pueden ocurrir (Test Coverage Items según ISO/IEC/IEEE 29119) y **Preparar** la Base de Datos de Prueba

Para estas tareas **Test4Data** proporciona **automatización** como **ayuda** al desarrollador y al tester



# (1) Herramientas



**SQLFpc/SQLRules** (web, ws / Java)

- Identifica Situaciones de interés a probar
- Evalúa cobertura respecto BD de pruebas

**QAGrow** (Java)

- Crea nuevas filas y las inserta en la BD de pruebas
- Para alcanzar la cobertura SQLFpc

**QAShrink** (Java)

- Inserta en la BD de pruebas filas de la BD de producción
- Para alcanzar la cobertura SQLFpc
- Minimizando el tamaño de la BD

# (2) Entrenamiento (SQLTest)

<http://in2test.lsi.uniovi.es/sqltest/>

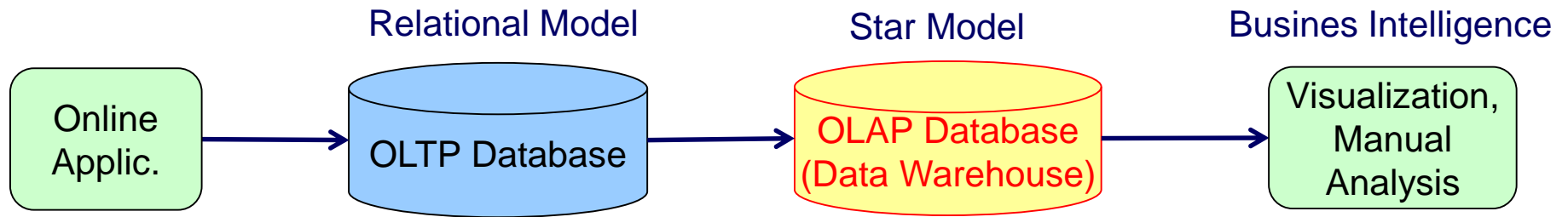
## ■ Objetivo: Realizar pruebas EFECTIVAS

- Facilidad para introducir datos de prueba y casos
- Evaluación automática de la efectividad de las pruebas
- Comparación con otros participantes y ranking

The screenshot displays the SQLTest web application interface, which is organized into several functional panels:

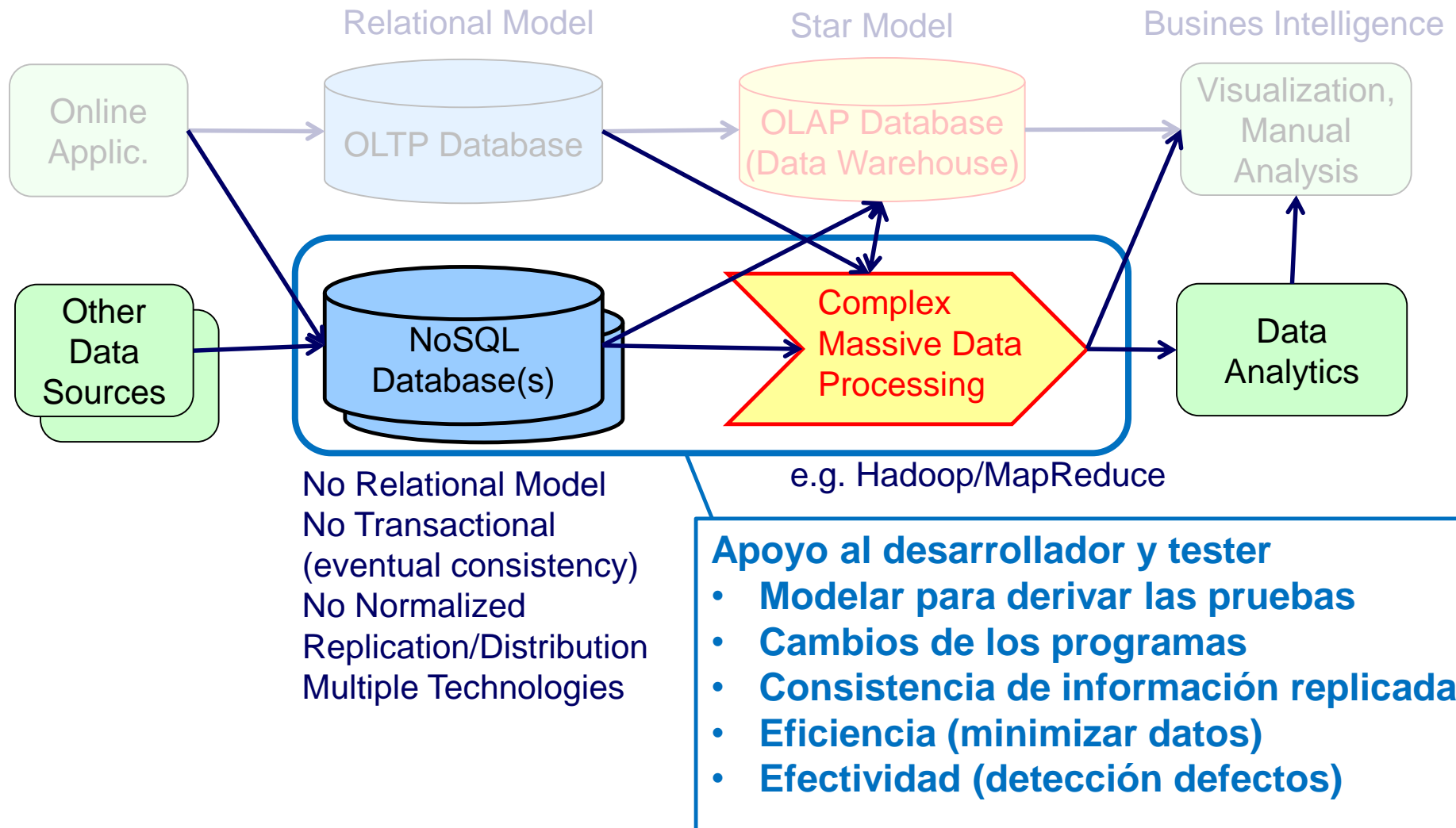
- Ejercicio:** A dropdown menu showing 'cvv14colisiones', 'cvv14iva2', and 'cvv14iva', with a 'RUN' button.
- Usuario:** A dropdown menu showing 'tuya', 'uo1', and 'uo1', with a 'RUN' button.
- Consulta:** A dropdown menu showing 'Q1', with 'RUN' and 'Eval' buttons.
- Grupo de pruebas:** A list of test groups including 'validacion', 'valor disponible', 'valor usado', and 'varios trimestres', with 'RUN', 'Add', 'Del', and 'Ren' buttons.
- Caso de prueba:** A dropdown menu showing 'faltareg', 'ultrim=4', and 'posnocompensa', with 'RUN', 'Add', and 'Del' buttons.
- Especificación:** A text area containing the problem description: 'Declaración de IVA trimestral. Ver la especificación del problema en el Campus Virtual'.
- Trimestres:** A table showing quarterly data for years 2000 and 2001.
- SQL a ejecutar antes del caso:** A text area containing the SQL statement: 'update trimestres set declarado=0,'.
- Entradas de usuario:** A text area containing the value '50'.
- Salida deseada:** A table showing the expected output for each quarter and year.
- Salida obtenida:** A table showing the actual output, which matches the expected output.
- Resultados y Mensajes de Error:** A text area showing a summary of results and a comparison with other participants.
- Comandos SQL Ejecutados:** A text area showing the SQL commands that were executed during the test.

# (3) Testing en Big Data Engineering

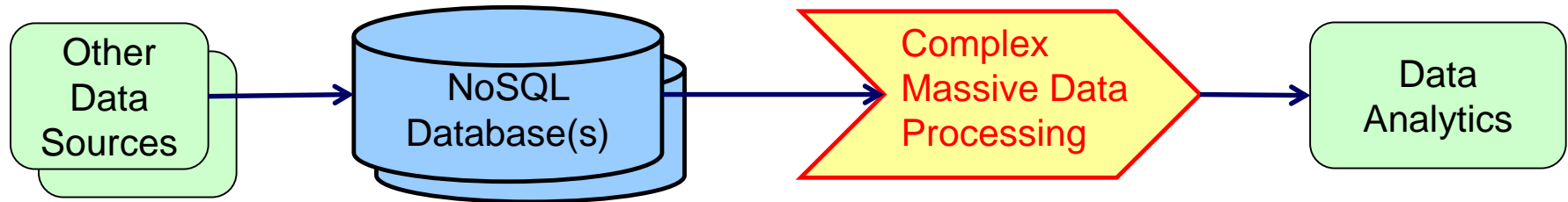




# (3) Testing en Big Data Engineering



# (3) Testing en Big Data Engineering



Investigación en curso:

Tipo almac.	Tecnología	Objetivo
Hadoop	MapReduce	Elaborar Técnicas de prueba: data-flow & fault-based Realizar pruebas aplicaciones mejora calidad datos
Columnas	Cassandra	Facilitar mantenimiento: detección impacto cambios al cambiar la estructura lógica de datos
Clave-Valor	Riak	Modelar y probar transacciones
Grafos	n/d	Evaluar grado de completitud de pruebas en aplicaciones ETL

# Evidencias: Usar SQLFpc para crear una BD de pruebas

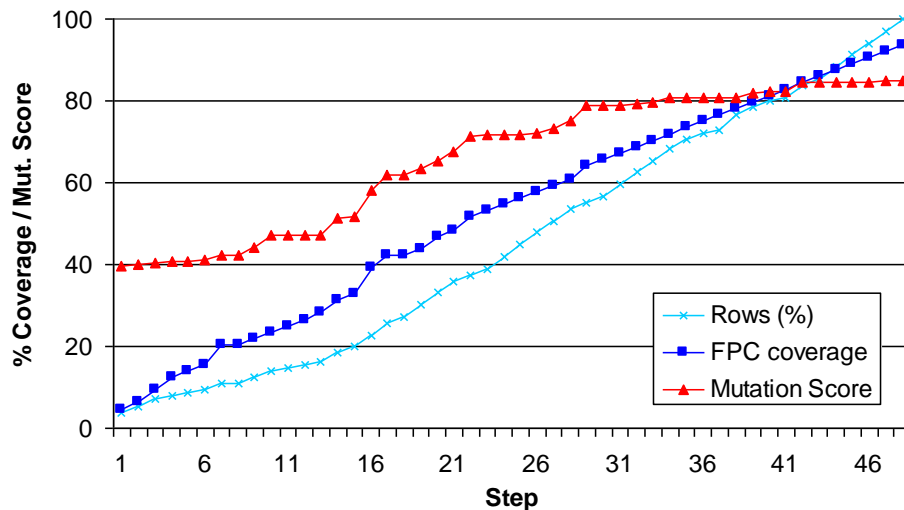
## ■ Proceso iterativo (ejemplo en ERP Compiere)

- SQLFpc proporciona reglas ejecutables (SQL) por cada situación a probar
- Ejecutar reglas no cubiertas
- Seleccionar una regla no cubierta
- Crear filas en BD para cubrir

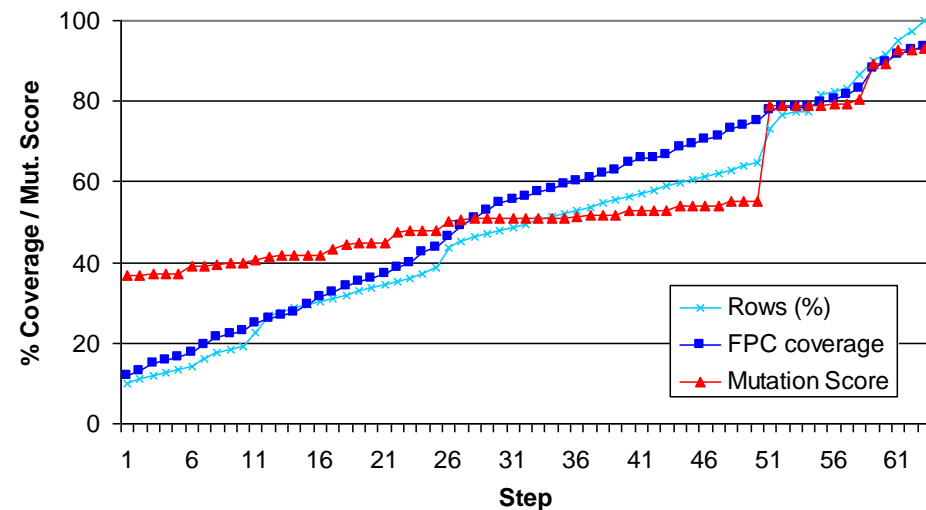
24 condiciones, 5 tablas → 64 filas

15 CASE, 20 tablas → 89 filas

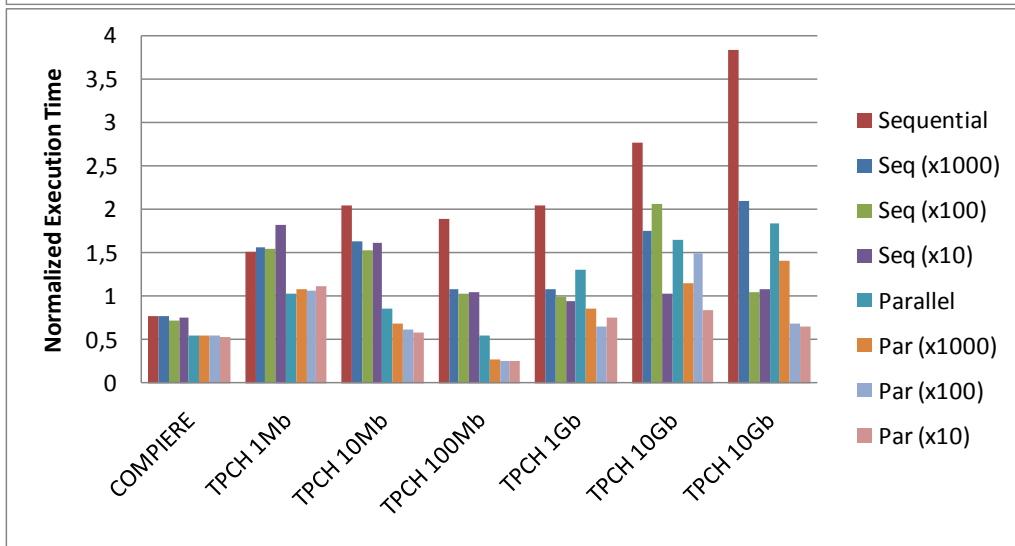
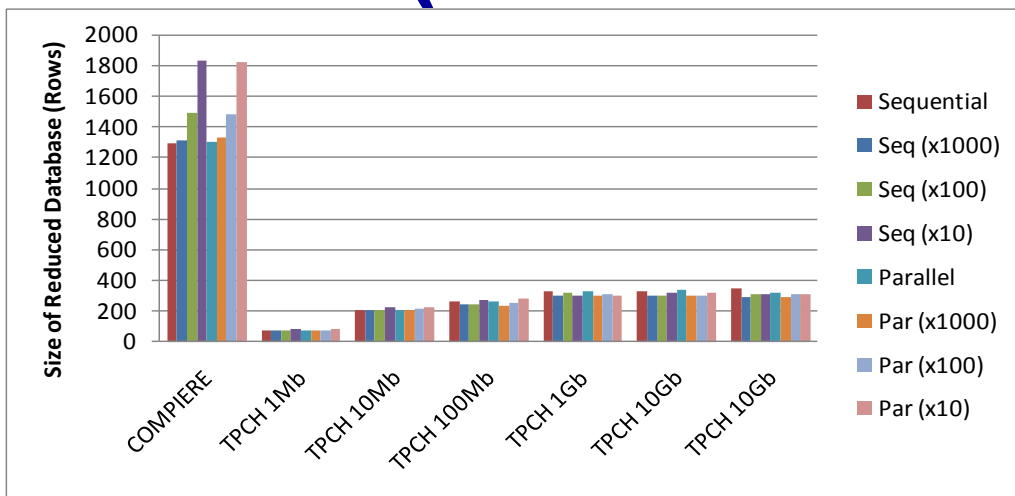
C\_Invoice\_Candidate\_v



C\_Invoice\_LineTax\_vt



# Evidencias: Reducción de una BD de 100GB (benchmark TPC-H)



- **Efectividad:** Preserva la cobertura respecto de la BD de producción
- **Eficiencia:** Reducción de tamaño independiente del de la BD de producción
- **Eficiencia:** Tiempo proporcional al de ejecución de las consultas
- **Funcionalidad:** Soporta BD grandes, queries complejas. Totalmente automatizado

# Cómo podemos ayudarnos mutuamente?

## ■ SQL - Test4Data

- Utilizarlas en la empresa
- Obtener feedback

## ■ NoSQL - Big Data Engineering

- Identificar problemática en las empresas
- Adaptación de nuestra investigación

## ■ Entrenamiento

- SQLTest
- Formación Técnicas de Prueba
- Divulgación ISO/IEC/IEEE 29119.  
Viabilidad aplicación/adaptación a PYMES

- Coste pruebas
- Tamaño BD pruebas
- + Significativa
- + Calidad y Cobertura
- + Productividad (automatización)
- Tareas manuales (creación BD)

# Contacto

- **GIIS** (Grupo de Investigación en Ingeniería del Software),  
Universidad Oviedo
  - <http://giis.uniovi.es/>
- Javier Tuya
  - <http://giis.uniovi.es/tuya/>
  - `tuya at uniovi.es`