

목적 : BenchGAD 프로젝트 사용할 클라이언트 프로그램의 gui 을 제작하기 위해 다양한 레퍼런스를 참고하고 그 자료를 정리하였음

## 목차

- 1.상용 데이터베이스 벤치마킹 툴 조사
- 2.논문에 기재된 벤치마킹 툴 조사
- 3.결론 - BenchGaD gui 방향

### 1.상용 데이터베이스 벤치마킹 툴 조사

- 1-1.Database Benchmark
- 1-2.HammerDB
- 1-3.DB Test Driven

#### 1-1.Database Benchmark



참고 : <https://github.com/STSSoft/DatabaseBenchmark>

#### 소개 :

Database Benchmark 는 대용량의 데이터와 함께 dbms 의 스트레스를 하는 오픈소스 툴 중 하나입니다. github 통해 관리하며 빠른 업데이트 피드백이 특징입니다.

#### 특징 :

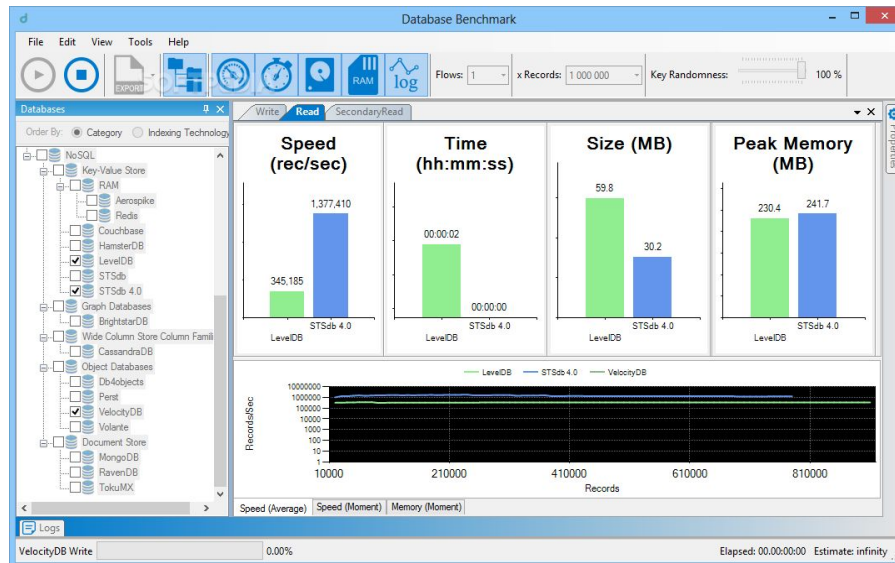
랜덤 혹은 순차적으로 생성된 대용량의 레코드를 삽입하여 테스트함

대용량의 레코드를 생성해주는 기능(랜덤 or 규칙적으로) 존재

output 을 시각화하여 보여주고 강력한 리포팅 옵션을 가지고 있습니다.

퍼포먼스 측정 툴보단 stress test 의 개념에 더 가까운 툴입니다.

## UI : 설정 창 및 output



web UI : [http://stssoft.com/benchmarks/?request\\_page=2](http://stssoft.com/benchmarks/?request_page=2)

Database benchmark 같은 경우 자사 홈페이지에서 툴의 기능을 제공해주고 있음

CATEGORIES		DATABASE NAME	FLOW X RECORD COUNT	RANDOM-NESS %	ELAPSED TIME	DATABASE SIZE	DATABASE INDEXING TECHNOLOGY
<input type="checkbox"/> Access 2013		<input type="checkbox"/> MongoDB	1x1000000	100	0h 02m 48s 535ms	165 MB	BTree
<input type="checkbox"/> Aerospike		<input type="checkbox"/> Firebird	1x50000	100	0h 01m 32s 969ms	7 MB	BTree
<input type="checkbox"/> Document Store		<input type="checkbox"/> MSSQL Server Compact 4.0	2x1000000	100	0h 33m 29s 564ms	96 MB	BTree
<input type="checkbox"/> Firebird		<input type="checkbox"/> Firebird	2x100	100	0h 00m 00s 259ms	1 MB	BTree
<input type="checkbox"/> Key-Value Store		<input type="checkbox"/> Oracle Berkeley DB	1x1000000	10	0h 00m 45s 211ms	94 MB	BTree
<input type="checkbox"/> LocalHost - Boletaria		<input type="checkbox"/> Firebird	1x1000000	10	0h 17m 52s	108 MB	BTree
<input type="checkbox"/> MonetDB							
<input type="checkbox"/> MongoDB							
<input type="checkbox"/> MSSQL Server 2012							
<input type="checkbox"/> MSSQL Server 2012 Clone							
<input type="checkbox"/> MSSQL Server Compact 4.0							
<input type="checkbox"/> MySQL (ARCHIVE)							
<input type="checkbox"/> MySQL (BLACKHOLE)							
<input type="checkbox"/> MySQL (INNODB)							
<input type="checkbox"/> MySQL (MyISAM)							
<input type="checkbox"/> MySQL (TokuDB)							
<input type="checkbox"/> NoSQL							
<input type="checkbox"/> Oracle Berkeley DB							
<input type="checkbox"/> Oracle GEISER							
<input type="checkbox"/> Postgre							
<input type="checkbox"/> Postgre Clone							
<input type="checkbox"/> RavenDB							
<input type="checkbox"/> Redis							
<input type="checkbox"/> SQL							

☐ 5,000,000

☐ 10,000,000

☐ 50,000,000

☐ 100,000,000

☐ 1,000,000,000

KEY TYPES

☐ Random

☐ Sequential

☐ Both

DATABASE SETTINGS

Submit

				375ms		
<input type="checkbox"/>	Firebird	2x50000	100	0h 00m 43s 514ms	7 MB	BTree
<input type="checkbox"/>	STSDb 4.0	1x300000	100	0h 00m 01s 358ms	8 MB	WaterfallTree
<input type="checkbox"/>	Firebird	1x1000000	100	0h 13m 05s 882ms	122 MB	BTree
<input type="checkbox"/>	Firebird	1x1000000	100	0h 10m 44s 047ms	122 MB	BTree
<input type="checkbox"/>	Firebird	1x1000000	100	0h 12m 12s 468ms	122 MB	BTree
<input type="checkbox"/>	MySQL (INNODB)	1x1000000	100	0h 01m 47s 892ms	112 MB	BTree

Compare selected

## 1-2. HammerDB



참고 : <http://www.hammerdb.com/document.html>  
<https://vimeo.com/174058490>

### 소개 :

HammerDB 는 오픈소스로 개발된 데이터베이스 테스트 및 벤치마킹 툴입니다.  
시중에서 별 다른 제약없이 사용 할 수 있는 툴이며 TPC-C TPC-H 등 최근 사용하는  
방식으로 구현한 툴입니다.

### 주요 기능:

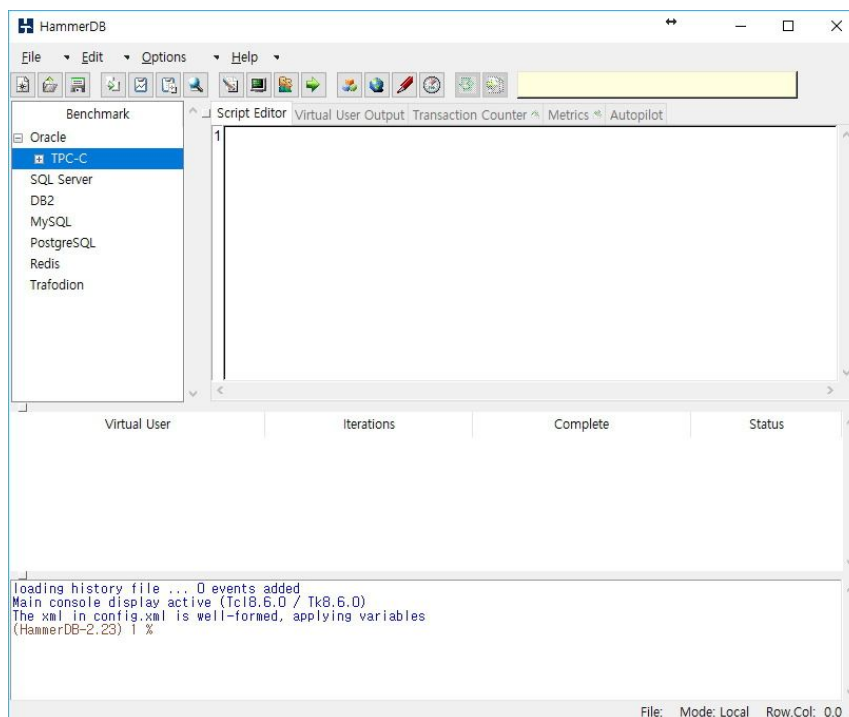
사용자들이 Gui 와 command line 을 통해 툴을 조작 가능

Linux 환경과 winodw 환경 모두다 정상 작동

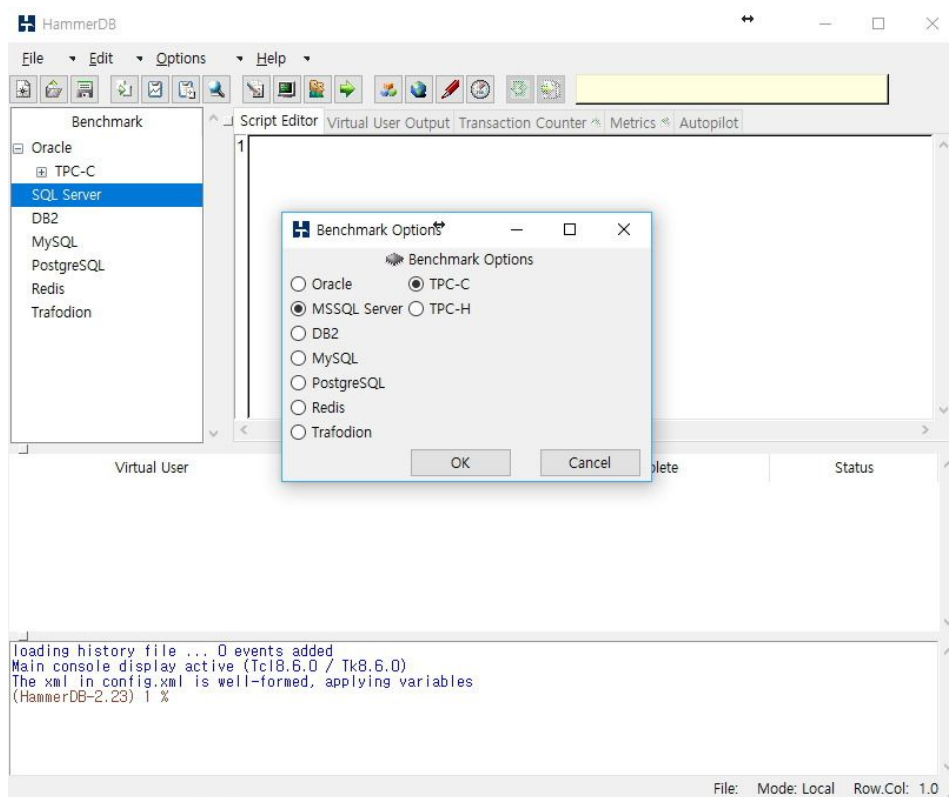
자동화 된 멀티 스레드 및 동적 스크립팅 지원을 통한 확장이 가능

### UI :

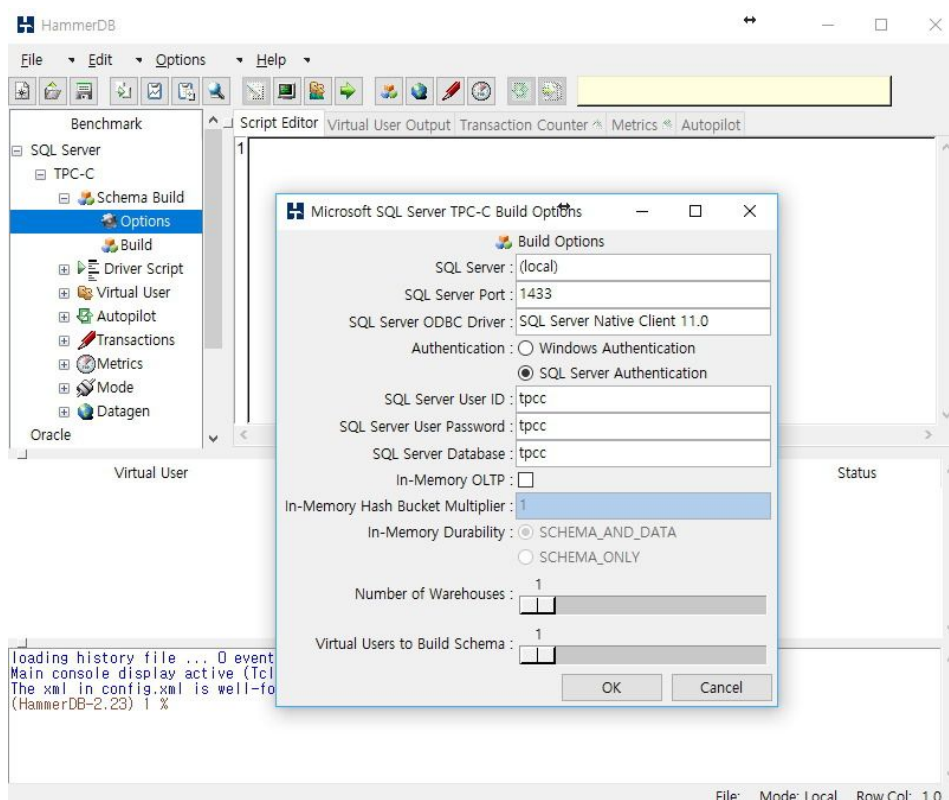
#### 시작 UI



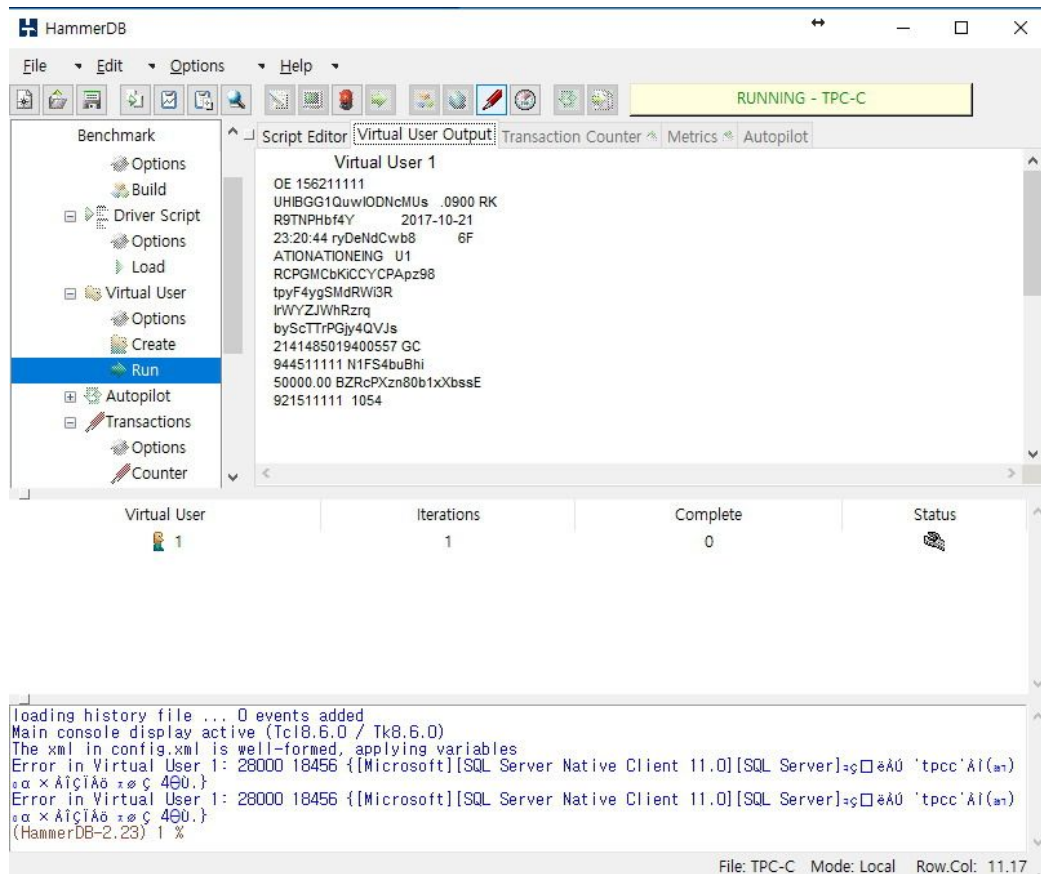
## DBMS 및 벤치마킹 방식 선택



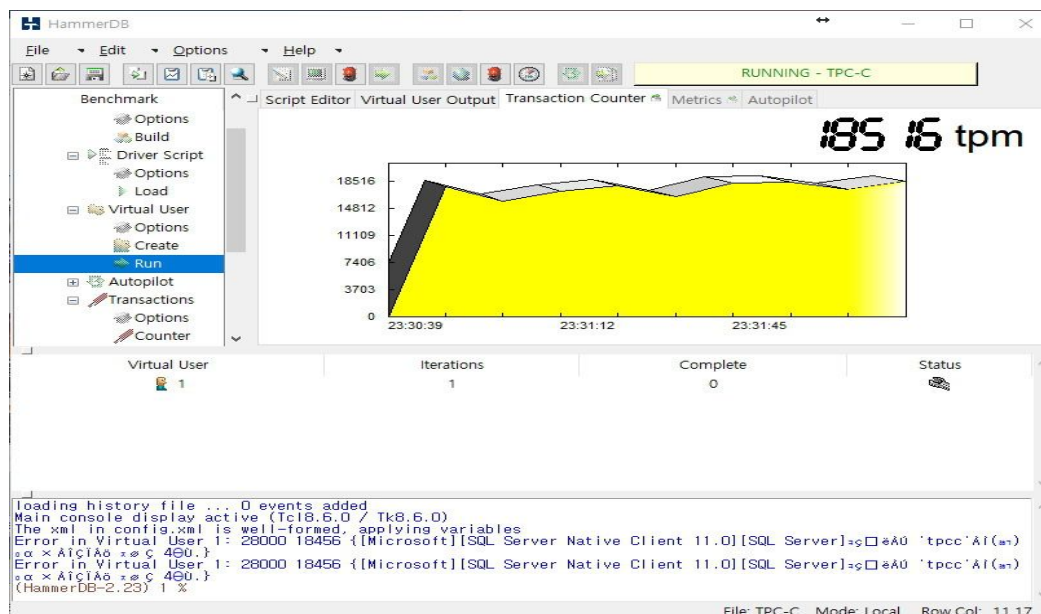
## 스키마 빌드 및 옵션 조정



## output 모습



## transaction 측정값





### 1-3. DB Test Driven

- 링크 : <http://www.dbtestdriven.com/>

SQLQuery1.sql - W...NT\apodlesny (57)\* x SQLServer-DbTestD...NT\apodlesny (59)

```
EXEC DBTD_Statistics @v_DatabaseName = 'DBTestDriven', @v_DisplayLevel = 'ALL';
```

(No column name)	StatName	StatValue	StatMetrics	SortOrder
1	#Statistics	Database Name	DBTestDriven	10
2	#Statistics	Code Coverage	99.4900000000 %	15
3	#Statistics	Number of Database objects	735	20
4	#Statistics	Number of objects Excluded from coverage analysis	547	30
5	#Statistics	Number of Tested objects	291	35
6	#Statistics	Number of Un-tested objects	1	40

(No column name)	Object Type	NumberOfObjects	NumberOfCoveredObjects	NumberOfNot TestedObjects	NumberOfExcept
1	#StatisticsByObject Type	DEFAULT_CONSTRAINT	9	0	0
2	#StatisticsByObject Type	SQL_SCALAR_FUNCTION	17	0	0
3	#StatisticsByObject Type	SQL_STORED_PROCEDURE	685	0	534
4	#StatisticsByObject Type	SQL_TABLE_VALUED_FUNCTION	1	0	0
5	#StatisticsByObject Type	SQL_TRIGGER	1	1	0
6	#StatisticsByObject Type	SYNONYM	2	0	1
7	#StatisticsByObject Type	USER_TABLE	17	13	4

(No column name)	ObjectID	ObjectName	ObjectDatabaseName	ObjectSchemaName	ObjectFullName
1	#AssertedObjects	NULL	DBTD_fnGetSuiteName_1234567890	DBTestDriven	dbo
2	#AssertedObjects	NULL	DBTD_fnGetSuiteName_ABC	DBTestDriven	dbo
3	#AssertedObjects	NULL	DBTD_sp_qwerty123456asdfg	DBTestDriven	dbo

(No column name)	ObjectID	ObjectName	ObjectDatabaseName	ObjectSchemaName	Object Type
1	#DatabaseObjects	1426104121	DF_DBTD_TBL_CODE_COVERAGE_EXCEPTION	dbo	DEFAULT_C
2	#DatabaseObjects	1586104691	DF_DBTD_TBL_LoadStatisticsTable_LoadTime	dbo	DEFAULT_C
3	#DatabaseObjects	1458104235	DF_DBTD_TBL_LOG_EventTime	dbo	DEFAULT_C

(No column name)	ObjectID	ObjectName	ObjectDatabaseName	ObjectSchemaName	Object Type
1	#UntestedObjects	NULL	tmp_DBTD_TMP_TRIGGER_TABLE_TRIGGER_A	dbo	SQL_TRIGGER

(No column name)	ObjectName	ObjectDatabaseName	ObjectSchemaName	Object Type	ExceptionDe	
1	#CoverageExceptions	tmp_DBTD_TEST_PROCEDURE_A	DBTestDriven	dbo	SQL_STORED_PROCEDURE	Unit Test H
2	#CoverageExceptions	UT_COMPATIBILITY_AssertColHasNoValue	DBTestDriven	dbo	SQL_STORED_PROCEDURE	Unit Test

(No column name)	ObjectID	ObjectName	ObjectDatabaseName	ObjectSchemaName	ObjectFullName
1	#CoverageDetails	NULL	DBTD_fnGetSuiteName_1234567890	DBTestDriven	dbo
2	#CoverageDetails	NULL	DBTD_fnGetSuiteName_ABC	DBTestDriven	dbo
3	#CoverageDetails	NULL	DBTD_fnGetSuiteName_ABC	DBTestDriven	dbo
4	#CoverageDetails	NULL	DBTD_sp_qwerty123456asdfg	DBTestDriven	dbo
5	#CoverageDetails	NULL	DBTD_sp_qwerty123456asdfg	DBTestDriven	dbo

Query executed successfully. WROC1059 (12.0 RTM) | FORTIGENT\apodlesny (57) | DBTestDriven | 00:00:00 | 1 rows

- 참고 자료

- <http://www.softwaretestingmagazine.com/tools/open-source-database-testing-tools/>

## 2.논문에 기재된 벤치마킹 툴 조사

### 2-1 SenseMark Abstraction

### 2-2 BenchGAD 프로젝트에서 수행 가능한 개선점

### 2-3 UI 분석

목적 :

시중에 나와있는 상용화 툴 이외에 논문에 기재되어있는 벤치마크 툴의 기능과 ui 를 벤치마킹을 한 뒤 BenchGaD project 에 사용 예정

### 2-1 SenseMark ( 논문에서 제작한 벤치마킹툴)

목적 : nosql dbms 시스템의 퍼포먼스를 측정하는데 의의를 둠

특징 : rest api와 wfs web services 사용하여 데이터를 전송 iot 기기를 통해 받은 다양한 형태의 데이터 값을 토대로 한 벤치마킹 시스템

### 2-2 BenchGAD 프로젝트에서 수행 가능한 개선점

시스템적 개선점 :

- 1) 단일 클러스터로 실행시키고 있다. ->gpu 를 사용
- 2) 아직 베타 버전인만큼 다양한 dbms를 지원하고 있지는 않다.-> 좀 더 많은 dbms지원

ui 적 개선점 :

- 1) 상용툴(hammer db 나 benchmark tool)과 비교하였을때 확실히 사용자에게 별로 친숙하지 않는 ui 나 직관적이지 않는 ui 를 보여준다.
  - 2) output 을 여러 자료형태로 보여주고 있지는 않다.
- > 처음 이 툴을 사용하는 사용자들에게도 메뉴얼에 의지하지 않고 툴을 사용할 수 있도록 직관적인 ui 개발 , output 값에 대한 다양한 정보 분석 제공( 차트, 시트 정보 제공 등등)



## 2-3 UI 분석

### SenseMark Architecture

## 6.2 Design of SenseMark

First step include to develop the **storage component** responsible for storing sensor data events (external data) and system component events (internal data) in Storage layer of sensing architecture[48]. The next step is to provide the **Adapter library** (Sensing layer) consisting of Java library for developing adapters capable of receiving data from all sensor types. Conceptual sensing architecture is shown in the figure 6.1.

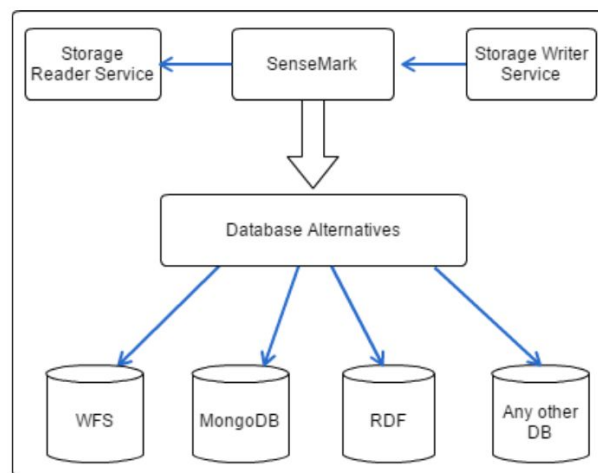


Figure 6.1: Conceptual Architecture of SenseMark

input 창 query를 직업 타이핑 하는 형식

SPARQL

Endpoint: [https://rdf.datagraft.net/4937384324/db/repositories/1604051091\\_measurement\\_june\\_sample\\_2016-processed](https://rdf.datagraft.net/4937384324/db/repositories/1604051091_measurement_june_sample_2016-processed)

Query

```
1: select *
2: where {
3: ?s ?p ?o . }
```

Figure 8.6: SPARQL Query

input 데이터 값에 대한 데이터 제공 ui 및 타입 지정

measurementid	fkey_observation	value	uom	observedproperty
3286379044	3275765003	0.69	ug/m3	PM1
3286379126	3275765010	24.14	C	Temperature
3286379127	3275765010	52.9	%	Relative Humidity
3286379128	3275765010	1.15	ug/m3	PM1
3286381068	3275767005	0	ug/m3	PM1
3286382073	3275768006	-30	ppb	NO
3286383090	3275769007	24.54	C	Temperature

Figure 8.2: RDF Mapping of CITI-SENSE CSV

clojure code 창 display

```

(def graph0 (prefix "http://www.citi-sense/"))

(def observation (prefix "http://citi-sense/vocab/observation/"))
(def rdf:type (rdf "type"))
(def Citi-Sense (prefix "http://citi-sense/vocab/"))
(def Citi-Sense:unitOfMeasurement (Citi-Sense "unitOfMeasurement"))
(def Citi-Sense:observedProperty (Citi-Sense "observedProperty"))
(def Citi-Sense:fkey (Citi-Sense "fkey"))
(def Citi-Sense:value (Citi-Sense "value"))
(def Citi-Sense:latitude (Citi-Sense "latitude"))
(def Citi-Sense:longitude (Citi-Sense "longitude"))
(def Citi-Sense:measureTime (Citi-Sense "measureTime"))
(def Citi-Sense:measureTimeUnix (Citi-Sense "measureTimeUnix"))
(def Citi-Sense:ugM3 (Citi-Sense "ugM3"))
(def Citi-Sense:caqiValue (Citi-Sense "caqiValue"))
(def Citi-Sense:caqiColor (Citi-Sense "caqiColor"))
(def Citi-Sense:geometry (Citi-Sense "geometry"))

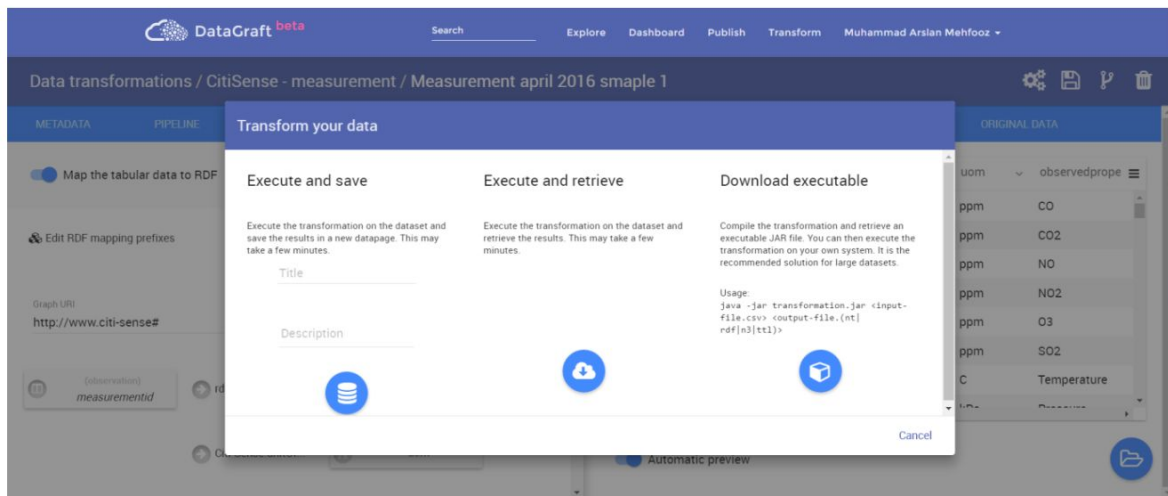
(defn double-literal [s] (if (nil? (re-matches (read-string "#\\[0-9.\\]+\\") s)) 0 (Double/parseDouble s)))
(defn integer-literal [s] (if (nil? (re-matches (read-string "#\\[0-9\\]+\\") s)) 0 (Integer/parseInt s)))

```

Figure 8.3: Overview of Clojure Code

excute 전 설정 창

For doing mapping larger CSV files to RDF, a Java based executable is downloaded in the system and mapping is performed locally on the machine to transform the data into RDF.



**Figure 8.4: Downloadable executable in DataCraft**

### 3.결론 - BenchGaD gui 방향

**직관적인 ui 사용** : 아이콘과 ui 위치 고려를 통한

Output 값 분석 : output. 값을 분석하여 사용자들에게 다양한 분석 결과를 제공 ( 도표화, 시트화)

**현재 작업 상황에 대한 정보 제공** : 프로그램 하단이나 상단바에 현재 진행상태(작업상황 퍼센티이지화 경과시간 등 )

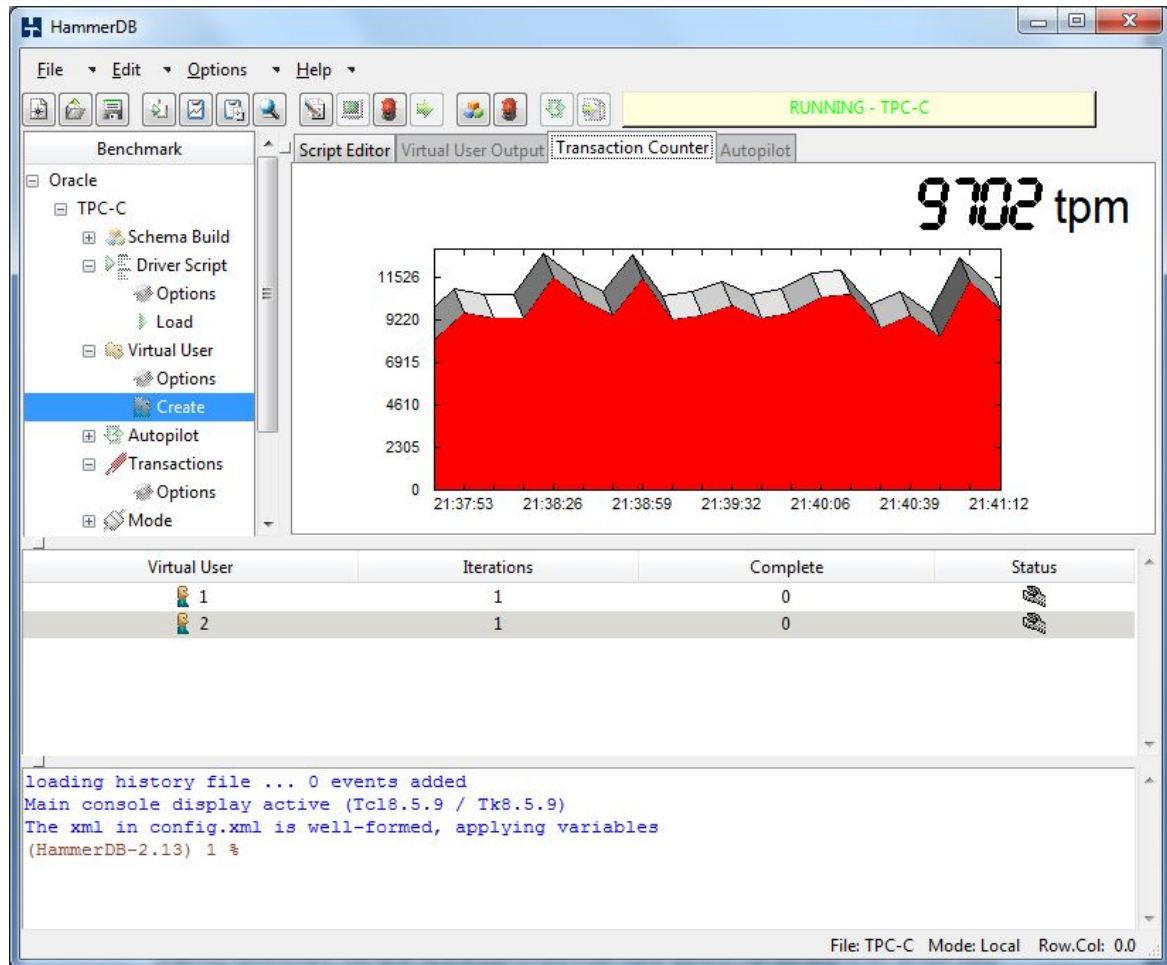
**진행 상황 컨트롤 ui 제공** : 정지, 일시정지, 재생 버튼을 통한 작업상황 컨트롤 사용자에게 제공 되도록이면 한 패널안에서 깔끔하게 제공 : 필요한 상황이 아니면 불필요한 프레임 호출은 피하는 방향(메모리적으로도 더 좋은 방향)

**현재 상황을 저장할 수 있는 ui.제공**( 현재는 환경설정 부분에서만 )

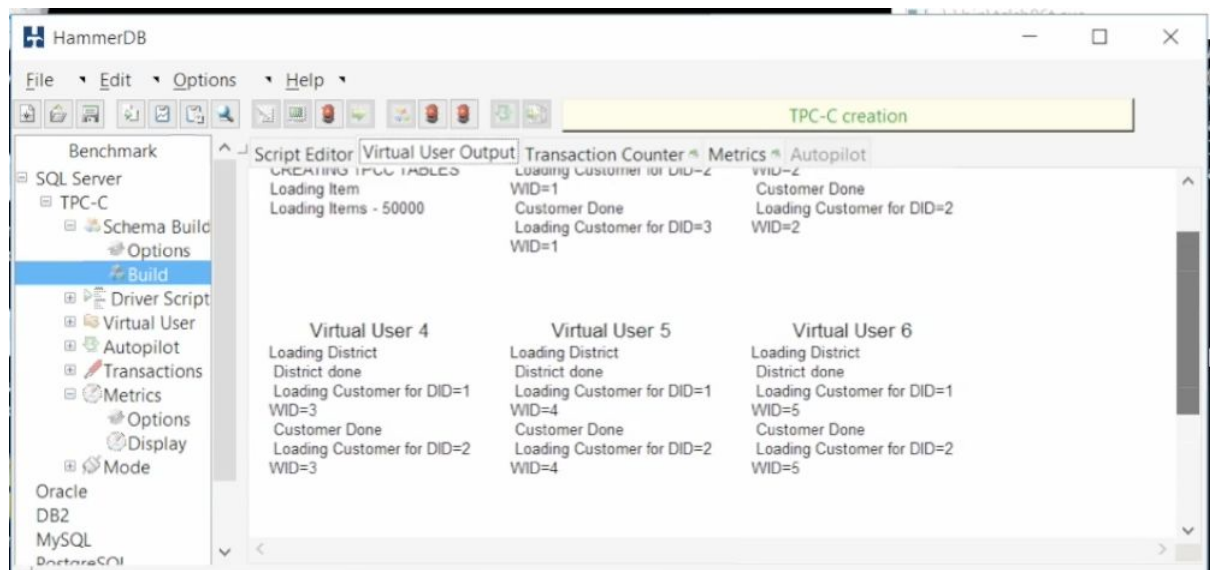
**또한 세팅정보를 로드할 수 있는 ui 제공** (마찬가지로 환경설정 부분에서만)

**Dbms 리스트를. Tree 개념 + 리스트 형식으로 직관적으로 제공**

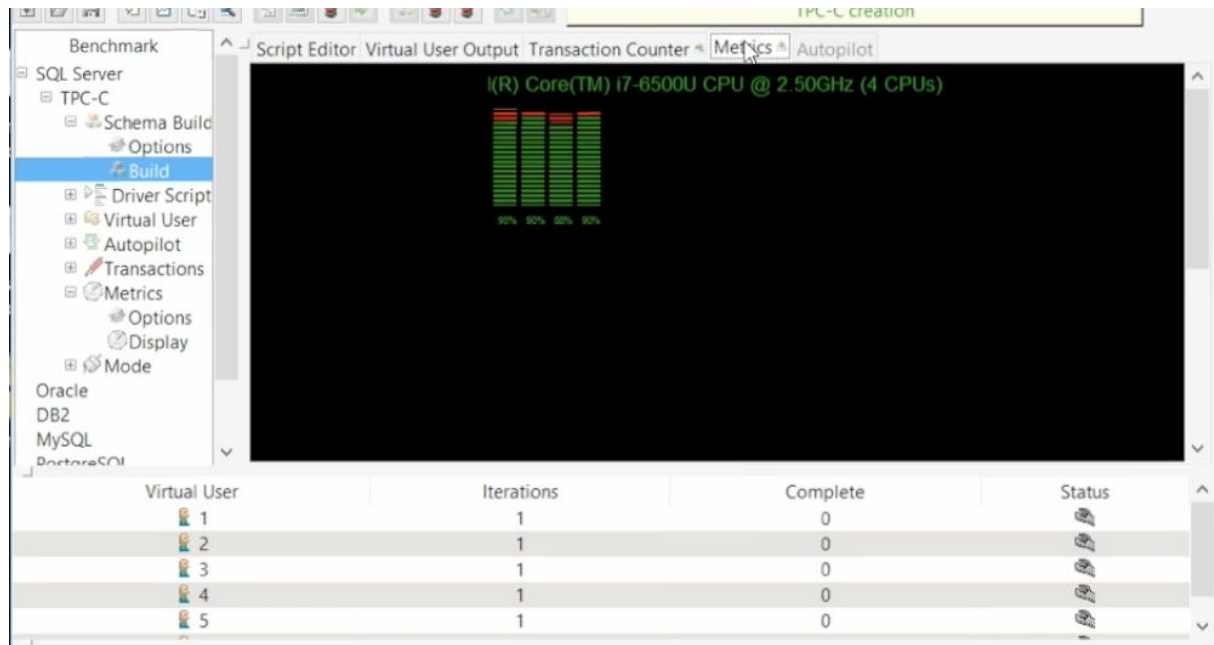
Output info 를 통해 만들것 :  
시간 추이에 따른 transaction counter



output 텍스트 형식으로 표현



## 시스템 모니터



결론 : 이 툴을 처음하는 사용자라도 메뉴얼에 의지하지 않고 스스로 사용할 수 있게 하는 방향으로 ui 를 제작할려함. 또한 상용화 툴에 못지 않는 ui 와 기능을 제공하도록 할것임