



PushToTest

Test Maker 5
Detailed tests results information system
Design document

1) Detailed database design

Database

The current database name is TMStatus. This database is accessed at the end of the test by a system in the tally and filled with the information of the test gathered in the xml files.

```
CREATE DATABASE `TMStatus`
```

Scenario

Scenario is the table in the database that contains the scenario information.

- S_ID is a key for the scenario.
- S_Name is the name of the scenario.
- S_Description is the description of the scenario.
- TS_Document is the Testscenario itself (The XML file)

```
CREATE TABLE Scenario (  
    S_ID INT(10) NOT NULL,  
    S_Name VARCHAR(50) NOT NULL,  
    S_Description VARCHAR(1000) NOT NULL,  
    TS_Document BLOB NOT NULL  
);  
ALTER TABLE Scenario ADD CONSTRAINT PRIMARY KEY (S_ID);
```

Resource

Resource is the table in the database that contains the resources of the test.

- S_ID is the number of the scenario.
- R_ID is the scenario's resource's number
- R_Resource is the actual resource (any file used: jar, text, xml) stored in the database.

The number of the scenario and the number of the resource inside the scenario makes the primary key

```
CREATE TABLE Resource (  
    R_ID INT(10) NOT NULL,  
    S_ID INT(10) NOT NULL,  
    R_Resource BLOB NOT NULL  
);  
ALTER TABLE Resource ADD CONSTRAINT PRIMARY KEY (R_ID, S_ID);
```

TestCase

TestCase is a table where the summary information of the testCase is stored

- S_ID is the number of the scenario.
- CVU_level is the level of the current virtual users
- Message_Index is the index of the message size.
- Case_name is the name of the case name
- TPS_Total the number of transactions per second of the test.

The S_ID plus the CVU_level plus the Message_Index form the primary key. If CVU_level and 1 is supposed, same happens with Message_Index.

```
CREATE TABLE TestCase (  
    S_ID INT(10) NOT NULL,  
    CVU_level INT(10) NOT NULL,  
    Message_Index INT(10) NOT NULL,  
    Case_name VARCHAR(20) NOT NULL,  
    TPS_Total DECIMAL(10,2) NOT NULL  
);  
ALTER TABLE TestCase ADD CONSTRAINT PRIMARY KEY (S_ID, CVU_level,  
Message_Index);
```

Host

Host is a table to contain the information of the hosts during the test. This table is mainly design to avoid recurrences of hosting information in the snapshot table

- S_ID is the number of the scenario.
- HOST_ID is the number of the host in the scenario
- HOST_URL is the url of the host
- HOST_NAME is name the of the host

```
CREATE TABLE Host (  
    S_ID INT(10) NOT NULL,  
    HOST_ID INT(10) NOT NULL,  
    HOST_URL VARCHAR(50) NOT NULL,  
    HOST_NAME VARCHAR(20) NOT NULL  
);  
ALTER TABLE Host ADD CONSTRAINT PRIMARY KEY (S_ID, HOST_ID);
```

Snapshot

Snapshot is a table that contains the information that every amount of time, one server sends to the console.

- Snapshot_ID is one unique identifier of the current snapshot
- S_ID is the number of the scenario.
- CVU_level is the level of the current virtual users
- Message_Index is the index of the message size.

- HOST_ID is the number of the host
- Snapshot_Start: Information where the snapshot was got by the console
- Snapshot_State State of the snapshot
 - initializing: making the warm up.
 - waiting: done making the warm up, waiting to start recording.
 - active: recording or making the teardown
 - complete: done with the teardown
- Snapshot_Average: the average transaction type in milliseconds
- Snapshot_Minimum: The minimum request time in milliseconds
- Snapshot_Maximum: The maximum request time in milliseconds
- Snapshot_Count: The number of transactions done

```
CREATE TABLE Snapshot (
    Snapshot_ID INT(10) NOT NULL,
    S_ID INT(10) NOT NULL,
    CVU_level INT(10) NOT NULL,
    Message_Index INT(10) NOT NULL,
    HOST_ID INT(10) NOT NULL,
    Snapshot_Start DATE NOT NULL,
    Snapshot_State VARCHAR(20) NOT NULL,
    Snapshot_Average DECIMAL(10,0) NOT NULL,
    Snapshot_Minimum DECIMAL(10,0) NOT NULL,
    Snapshot_Maximum DECIMAL(10,0) NOT NULL,
    Snapshot_Count INT(10) NOT NULL
);
ALTER TABLE Snapshot ADD CONSTRAINT PRIMARY KEY (Snapshot_ID);
```

Monitor

Monitor is a table that contains the information that every amount of time, one monitor sends to the console. The monitor can be a PTTMonitor or and SNMP monitor.

- Monitor_ID is one unique identifier of the current monitor
- S_ID is the number of the scenario.
- CVU_level is the level of the current virtual users
- Message_Index is the index of the message size.
- HOST_ID is the number of the host
- Count is the number of ticks of the monitor

```
CREATE TABLE TMStatus.Monitor (
    Monitor_ID INT(10) NOT NULL,
    S_ID INT(10) NOT NULL,
    CVU_level INT(10) NOT NULL,
    Message_Index INT(10) NOT NULL,
    HOST_ID INT(10) NOT NULL,
    count INT(10) NOT NULL
```

```
);  
ALTER TABLE TMStatus.Monitor ADD CONSTRAINT PRIMARY KEY  
(Monitor_ID);
```

Monitor_Data

Monitor data contains each one of the ticks given by the monitor during its time. The type field can be CPU, MEM or NET. Per_Use is the percentage of that resource used during the sample.

- Monitor_ID is the ID of the monitor
- Row_ID is the number of the row in the monitor
- Per_Use is the number of the usage of the resource
- Type is the type of the resource
- Time is the moment when the sample was taken.

```
CREATE TABLE Monitor_Data (  
    Row_ID INT(10) NOT NULL,  
    Monitor_ID INT(10) NOT NULL,  
    Per_Use DECIMAL(2,2) NOT NULL,  
    Type VARCHAR(3) NOT NULL,  
    Time TIME NOT NULL  
);
```

```
ALTER TABLE Monitor_Data ADD CONSTRAINT PRIMARY KEY (Row_ID,  
Monitor_ID);
```

Transactions

“Transactions” is a table that contains the information that the Test-nodes send to the console

- T_ID is an unique identifier for the current transaction
- S_ID is the number of the scenario.
- CVU_level is the level of the current virtual users
- Message_Index is the index of the message size.
- HOST_ID is the number of the host

```
CREATE TABLE Transactions (  
    T_ID INT(10) NOT NULL,  
    S_ID INT(10) NOT NULL,  
    HOST_ID INT(10) NOT NULL,  
    CVU_level INT(10) NOT NULL,  
    Message_Index INT(10) NOT NULL  
);
```

```
ALTER TABLE Transactions ADD CONSTRAINT PRIMARY KEY (T_ID);
```

Transactions_Data

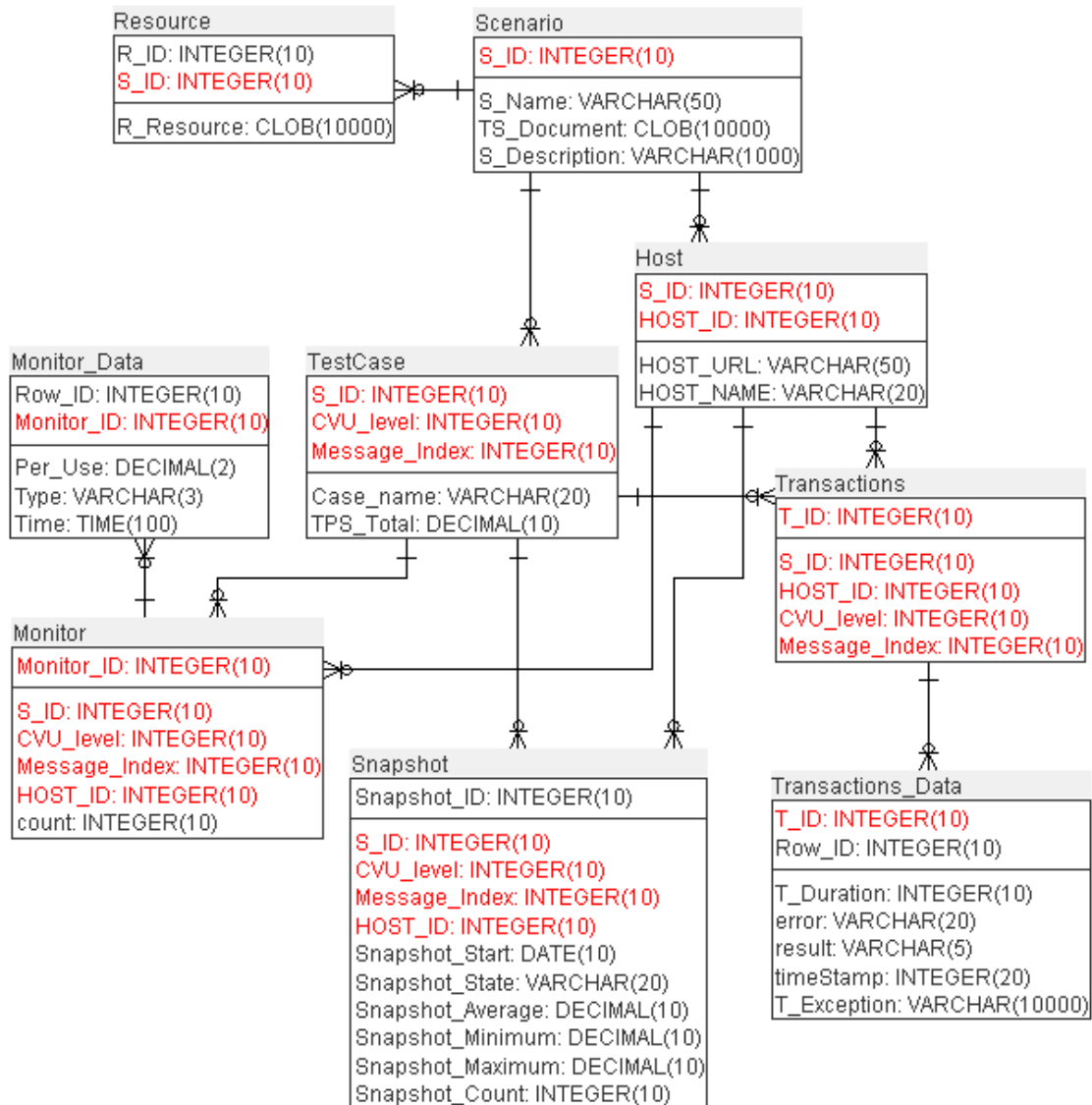
Transactions_Data is a table that contains all the information about the transactions done and the results of the transactions.

- T_ID is the identifier of the transaction
- Row_ID is the number of the transaction row.
- T_Duration is the duration of the transaction in milliseconds
- Error will show the error message of the transaction if it fails.
- Result true if the transaction succeed, false otherwise.
- Timestamp. Returns the number of milliseconds since January 1, 1970, 00:00:00 GMT (Java new Date().getTime());
- T_Exception: Is the stack trace of the exception is the transaction fails.

```
CREATE TABLE Transactions (  
    T_ID INT(10) NOT NULL,  
    S_ID INT(10) NOT NULL,  
    HOST_ID INT(10) NOT NULL,  
    CVU_level INT(10) NOT NULL,  
    Message_Index INT(10) NOT NULL  
);  
ALTER TABLE Transactions ADD CONSTRAINT PRIMARY KEY (T_ID);
```

Database Constraint

The next picture shows the current database foreign keys constraints



Constrain in SQL

```
ALTER TABLE Resource ADD CONSTRAINT Scenario_Resource_fk  
FOREIGN KEY (S_ID)  
REFERENCES Scenario (S_ID);
```

```
ALTER TABLE TestCase ADD CONSTRAINT Scenario_TestCase_fk  
FOREIGN KEY (S_ID)  
REFERENCES Scenario (S_ID);
```

```
ALTER TABLE Host ADD CONSTRAINT Scenario_New_Table_fk
FOREIGN KEY (S_ID)
REFERENCES Scenario (S_ID);
```

```
ALTER TABLE Snapshot ADD CONSTRAINT TestCase_Case_Exec_Snapshot_fk
FOREIGN KEY (CVU_level, Message_Index, S_ID)
REFERENCES TestCase (CVU_level, Message_Index, S_ID);
```

```
ALTER TABLE Monitor ADD CONSTRAINT TestCase_Monitor_fk
FOREIGN KEY (S_ID, CVU_level, Message_Index)
REFERENCES TestCase (S_ID, CVU_level, Message_Index);
```

```
ALTER TABLE Transactions ADD CONSTRAINT TestCase_Transactions_fk
FOREIGN KEY (S_ID, CVU_level, Message_Index)
REFERENCES TestCase (S_ID, CVU_level, Message_Index);
```

```
ALTER TABLE Monitor_Data ADD CONSTRAINT Monitor_Monitor_Data_fk
FOREIGN KEY (Monitor_ID)
REFERENCES Monitor (Monitor_ID);
```

```
ALTER TABLE Snapshot ADD CONSTRAINT New_Table_Snapshot_fk
FOREIGN KEY (S_ID, HOST_ID)
REFERENCES Host (S_ID, HOST_ID);
```

```
ALTER TABLE Monitor ADD CONSTRAINT Host_Monitor_fk
FOREIGN KEY (S_ID, HOST_ID)
REFERENCES Host (S_ID, HOST_ID);
```

```
ALTER TABLE Transactions ADD CONSTRAINT Host_Transactions_fk
FOREIGN KEY (S_ID, HOST_ID)
REFERENCES Host (S_ID, HOST_ID);
```

```
ALTER TABLE Transactions_Data ADD CONSTRAINT
Transactions_Transactions_Data_fk
FOREIGN KEY (T_ID)
REFERENCES Transactions (T_ID);
```