Data Generator For SAP Solutions Using Benerator

Baba Pakruddin Tailor
Julian Reddy Allam
Naga Sai Krishna Ayinampudi
Sai Rajesh Vanimireddy

OTTO-VON-GUERICKE UNIVERSITY MAGDEBURG,GERMANY

SAP UCC MAGDEBURG

This project is done under the supervision of Mr.Tim and Mr.Chris,SAP UCC Magdeburg .

2016

Contents

1. Introduction

1.1 Project overview

Getting realistic data for analysis is tedious because of privacy and access restrictions. The purpose of the project is to customize and configure the Benerator, open source tool for SAP solutions (ERP, CRM, etc.) to create realistic data sets. The Business processes of SAP ERP and the data types have been considered in the data generation process. The generated datasets were used for analysis.

1.2 Scope and objective

- Customizing and configuring the benerator tool
- Evaluate existing historical data of GBI
- Find resourceful insights with Pattern recognition
- Huge data sets for analysis purposes in SAP BW and SAP HANA
- Research possibilities “with Benerator”.

1.3 Assignment of Roles and Responsibilities

Team Leader: Baba Pakruddin Tailor
Programmer: Baba Pakruddin Tailor, Nagasai Krishna Ayinampudi
Tester : Julian Reddy Allam, Sai Rajesh Vanimireddy
Documentation: Baba Pakruddin Tailor , Julian Reddy Allam
Scientific Paper: Sai Rajesh Vanimireddy

1.4 Project schedule

Task	Responsible Person	Completion Date
Literature Search	Baba Pakruddin,Naga Sai	09/02/2015
Related Work	Baba Pakruddin, Nagasai	9/25/2015
Concept Development	Baba Pakruddin, Nagasai	01/12/2016
Configuring Benerator	Baba Pakruddin	04/25/2016
Testing	Julian Allam, Sai Rajesh	05/10/2016
Analysis	Julian Allam, Sai Rajesh	06/4/2016
Documentation	Julian Allam	06/20/2016
Scientific Paper	Sai Rajesh	06/20/2016
Project completion		09/02/2016

2. Product Description

2.1 Product Context

Benerator is an open source tool which has powerful capability to generate data in different formats, that would best suit the requirements for Data generation for SAP Solutions. It supports all data types of My SQL Server, Oracle, DB2, My SQL 5 and HSQL1.8.

2.2 Dependencies

Server Requirements: • Secure WAMP is an open source Web server hosting application. It mainly provides four important elements for web server which are database, web scripting software, an operating system and a web server.

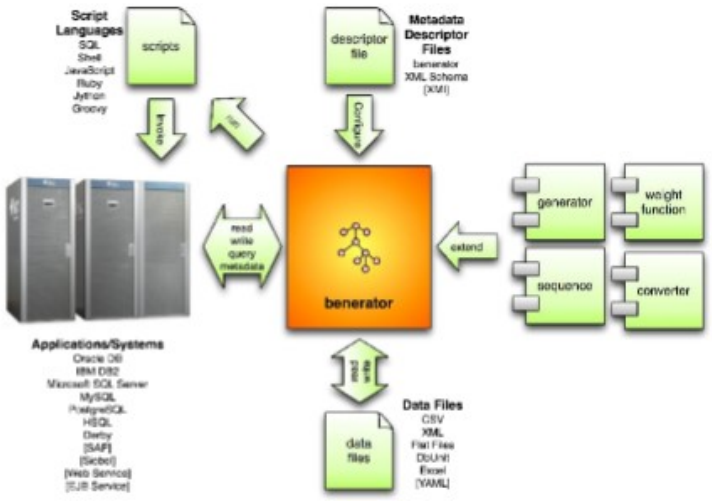
• MySQL Workbench is a visual database design tool that integrates SQL development, administration, creation and maintenance into a single integrated development environment for the MySQL database system.

• Benerator is a tool for generating realistic and a high volume test data for the system under test. It also reduces the time during creation of data setup of xml Schema or any enterprise application.

2.3 Benerator Block diagram and its Goals

Goals of Benerator

1. Efficiently generating the huge data volumes
2. Intuitive data definition format
3. Establishing a common data generation platform for software systems and business domain.
4. Data generation that satisfies complex data validity requirements.
5. Easily accessible by non-developers.


Script Languages
SQL
Shell
JavaScript
Ruby
Jython
Groovy
scripts
invoke
descriptor file
Metadata Descriptor Files
benerator
XML Schema
(XML)
configure
generator
weight function
extend
sequence
converter
read
write
query
metadata
benerator
Applications/Systems
Oracle DB
IBM DB2
Microsoft SQL Server
MySQL
PostgreSQL
HSQL
Derby
[SAP]
[Siebel]
[Web Service]
[EJB Service]
data files
Data Files
CSV
XML
Flat Files
DbUnit
Excel
[YAML]

3. Requirements

3.1 Functional Requirements

Master Data It is a unique business data used across multiple systems or applications. It is a core data necessary for operations in any business units. Generally, types of information treated as Master data varies from company to company or department to department within the company. Master data is highly secured and non-transactional in nature. So using this data, we can create a referential data in order to modify according to the developer requirements. In our project we have used Master data of Customer, Vendor and Product to create a realistic data.

Example of Master data for the Vendor, Product and Customer are shown below

3.1.1 Master Data of vendor

AUGSBURG	000	LOHSE SCHRAUBE	118000	DE00
BIELEFELD	000	PYRAMID BIKING	114000	DE00
BRAUNSCHWEIG	000	BURGMEISTER ZUBEHOR C	113000	DE00
DRESDEN	000	SACHSEN STAHL AG	123000	DE00
ERFURT	000	ABS BRAKES GMBH	115000	DE00
HEILBRONN	000	THICK SPOKE	119000	DE00
KARLSRUHE	000	GUMMI SCHULTZE	117000	DE00
KOLN	000	COLOGNE BIKE SUPPLIES	122000	DE00
MANNHEIM	000	FLAT TIRE AND MORE	116000	DE00
MUNCHEN	000	RUN & FUN	124000	DE00
SANDKRUG	000	SHELL GEAR	121000	DE00
WURZBURG	000	MAIN CARBON	120000	DE00

Figure 3.1: Vendor DE

ATLANTA	000	OLYMPIC PROTECTIVE GEAR	101000	US00
CARLSBAD	000	REDWOOD KITS	112000	US00
CINCINNATI	000	BOOMTOWN TIRE &WHEEL	102000	US00
EDISON	000	LIGHTBULB ACCESSORY KITS	104000	US00
GRAND RAPIDS	000	RAPIDS NUTS N BOLTS	108000	US00
HOUSTON	000	SPACE BIKE COMPOSITES	105000	US00
IRVING	000	DALLAS BIKE BASICS	103000	US00
LACROSSE	000	NIGHT RIDER ALUMINUM PROD	106000	US00
MCLEAN	000	SPY GEAR	107000	US00
MIAMI	000	FUN N THE SUN SEATS N BAR	110000	US00
PORTLAND	000	GREEN BLAZERS SEATS	109000	US00
SCOTTSDALE	000	SUNNY SIDE UP TIRE	111000	US00

Figure 3.2: Vendor US

3.1.2 Master Data of Product

CARBON COMPOSITWHEEL ASSEMBLY	EN	CCWA1000
OFF ROAD ALUMINUM WHEEL	EN	ALWA1000
TOURING ALUMINUM WHEEL ASSEMBLY	EN	ALWA2000

Figure 3.3: productsemi

Accessories		
AIR PUMP	EN	PUMP1000
ELBOW PADS	EN	EPAD1000
FIRST AID KIT	EN	FAID1000
KNEE PADS	EN	KPAD1000
OFF ROAD HELMET	EN	OHMT1000
REPAIR KIT	EN	RKIT1000
ROAD HELMET	EN	RHMT1000
T-SHIRT	EN	SHRT1000
WATER BOTTLE	EN	BOTL1000
WATER BOTTLE CAGE	EN	CAGE1000

Figure 3.4: prodtrading

3.1.3 Master Data of Customer

3.1.4 Business Processes

- Procure-to-Pay
- Order-to-Cash

Touring Bikes (Deluxe, Professional) in three colors		
Off-Road Bikes (Men, Women)		
DELUXE TOURING BIKE (BLACK)	EN	DXTR1000
DELUXE TOURING BIKE (RED)	EN	DXTR3000
DELUXE TOURING BIKE (SILVER)	EN	DXTR2000
MEN'S OFF ROAD BIKE	EN	ORMN1000
PROFESSIONAL TOURING BIKE (BLACK)	EN	PRTR1000
PROFESSIONAL TOURING BIKE (RED)	EN	PRTR3000
PROFESSIONAL TOURING BIKE (SILVER)	EN	PRTR2000
WOMEN'S OFF ROAD BIKE EN	EN	ORWN1000

Figure 3.5: productfinished

Procure-to-Pay is the process of obtaining and managing the raw materials needed for manufacturing a product. This Process involves the activities of requesting, purchasing, receiving, paying and accounting for the goods/services.

Procure-to-Pay Process steps – SAP modules involved - MM, FI

Steps	Process
1	New vendor
2	Material master for trading goods
3	Extend Material master for trading goods
4	Display stock requirements list
5	Create Purchase requisition
6	Display Stock requirements list
7	Create Request for quotation
8	Maintain quotations from vendors
9	Evaluate quotations on price
10	Reject quotations
11	Create purchase order referencing an RFQ
12	Display purchase order
13	Create goods Receipt for Purchase order
14	Verify physical receipt of goods
15	Create invoice receipt from vendor
16	Create goods receipt for Purchase order
17	Create invoice receipt from vendor
18	Post payments to vendor
19	Display vendor line items
20	Display G/L account balances and line items

Order-to-Cash is a set of business processes that involve receiving and fulfilling customer requests for goods or services. SAP modules involved – SD, MM, FI

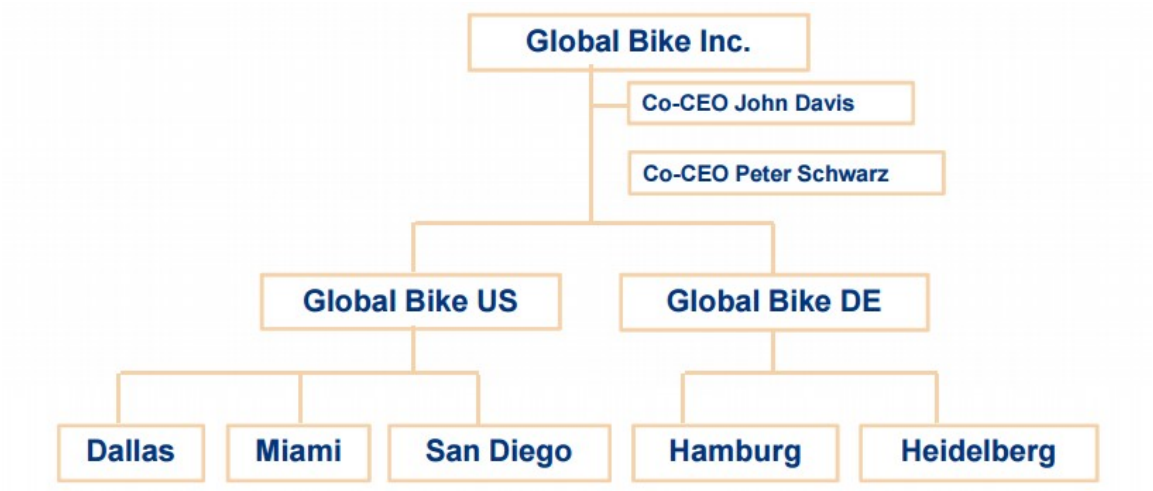
Steps	Process
1	New Customer
2	Create Contact Person for customer
3	Change customer
4	Create Customer Inquiry
5	Create Customer Quotation
6	Sales order referencing a quotation
7	Stock Status
8	Display Sales order
9	Start Delivery Process
10	Stock Status
11	Pick materials on delivery notes
12	Post Goods issue
13	Stock Status
14	Invoice for customer
15	Display billing Doc Invoice
16	Post receipt of customer payment
17	Review Doc Flow

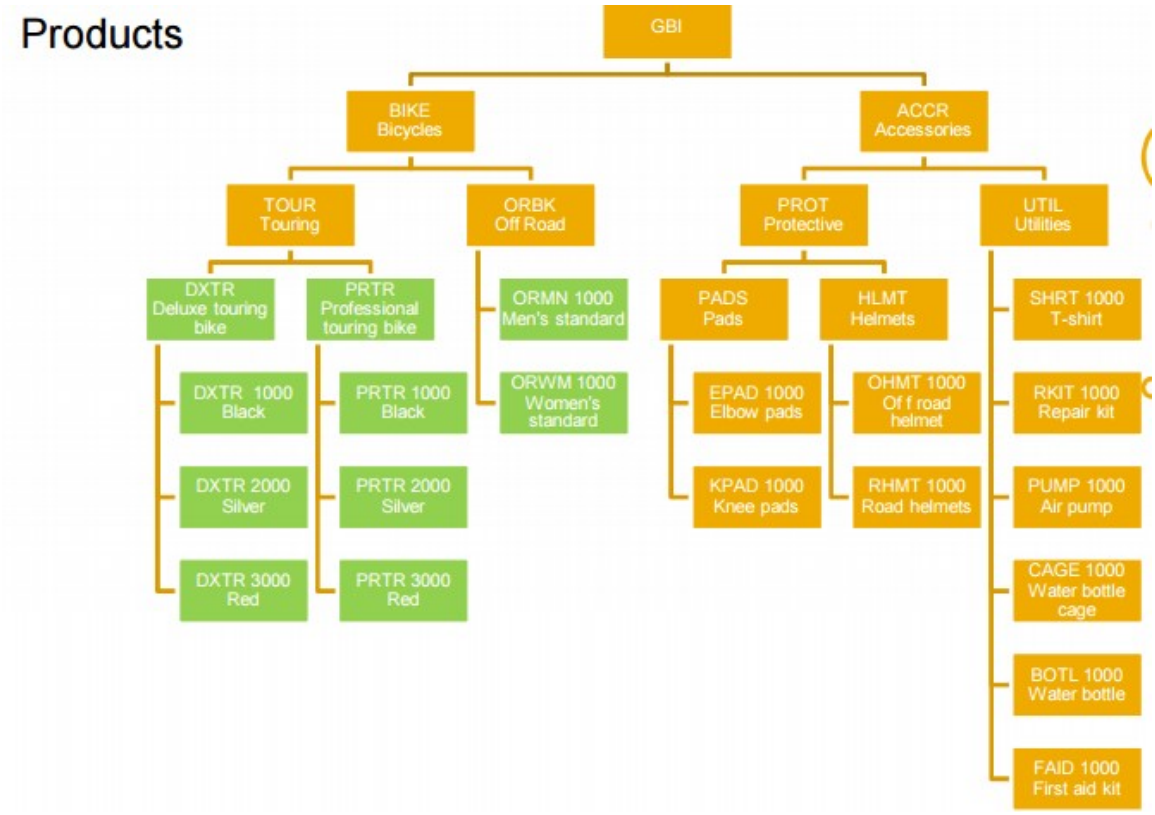
4. Design Analysis

4.1 Global Bike Inc

Global Bike Inc. is a world class bicycle company serving the professional cyclists for touring and off-road racing. John Davis and Peter Schwartz together established GBI Company in Germany and in US. It implements SAP ERP with all ERP functions being centralized. Its main objective is to deliver low-cost bikes and best-in-class technology to all divisions globally. The organization has the Business processes namely Order-to-cash, Procure-to-Pay, Idea-to-Market, Build-to-Stock, Service and Support.

4.1.1 organizational structure of GBI





4.1.2 Products of GBI

4.2 OLAP

OLAP (Online analytical processing) is a concept that describes an approach to answer multi-dimensional analytical queries. It is the key for different kinds of business performance management, forecasting, planning, analysis, knowledge discovery.

4.3 Data Types

We do analysis to support decision-making. Data forms the basis for analysis and is generated using benerator tool. The target SAP BW system which is used for analysis of data contains a multi-dimensional (star schema) data container which is called as Infocube. Infocube has two data types namely Characteristics and Key figures. Characteristics are the business objects or the business entity. For example customer, product etc. Whereas, Key figures are the quantitative numbers of the business entity or business process. Business process will consists of Master data of Customers, Products, Vendors and system functions in SAP. By Considering the Master data of GBI, we have selected the data required for a complete Business Process with a focus on Order-to-Cash and Procure-to-Pay business processes. Benerator will generate following data types that are part of the Business processes as shown below.

S.no	Entry	Format	Example	M
1	Customer	Numeric	0000001000	
2	Contact Person	Numeric	0000000001	
3	Company code	Alphanumeric	US00 for US-DE00 for Germany	For US compani
4	Customer Inquiry	Numeric	0010000001	
5	Customer Quotation	Numeric	0020000001	
6	Sales-order referencing quotation	Numeric	0000000001	
7	Outbound Delivery	Numeric	0080000001	
8	Invoice for customer	Numeric	0090000001	
9	Post Receipt of Customer Payment	Numeric	1400000001	
10	Vendor	Numeric	0000101000	
11	Material master for trading goods	Alphanumeric	PUMP1000	
12	Purchase requisition	Numeric	0010000000	
13	Request for quotation	Numeric	6000000000	
14	Purchase order referencing an RFQ	Numeric	4500000001	
15	Goods Receipt for Purchase order	Numeric	5000000001	
16	Invoice Receipt from vendor	Numeric	5105600101	
17	Goods Receipt for Purchase order	Numeric	5000000002	
18	Post Payments to Vendor	Numeric	1500000001	
19	Receive Goods from Production Order	Numeric	5000000011	

4.4 Analyes-Scenario

Sales are the backbone of any company or organisation in business processes. Analyses done for the generated data sales in a business organisations will help to known the sales in a business.This helps in making the betterment in sales if there is a decrement or fall in sales. It also helps to known the future prediction sales.

In our Benerator tool to make the generated data to be realistic,we followed following rules.

- Region-wise customer distribution
- Seasonal-wise customer distribution and
- Product preferences

5. TOOLS INSTALLATION

5.1 Tools

We have used three tools to generate the data
Benerator Tool
Secure WAMP
My SQL workbench

5.2 Process for Tools installation

- To install Secure WAMP
- Click on the link to download and install using http://securewamp.org/en/

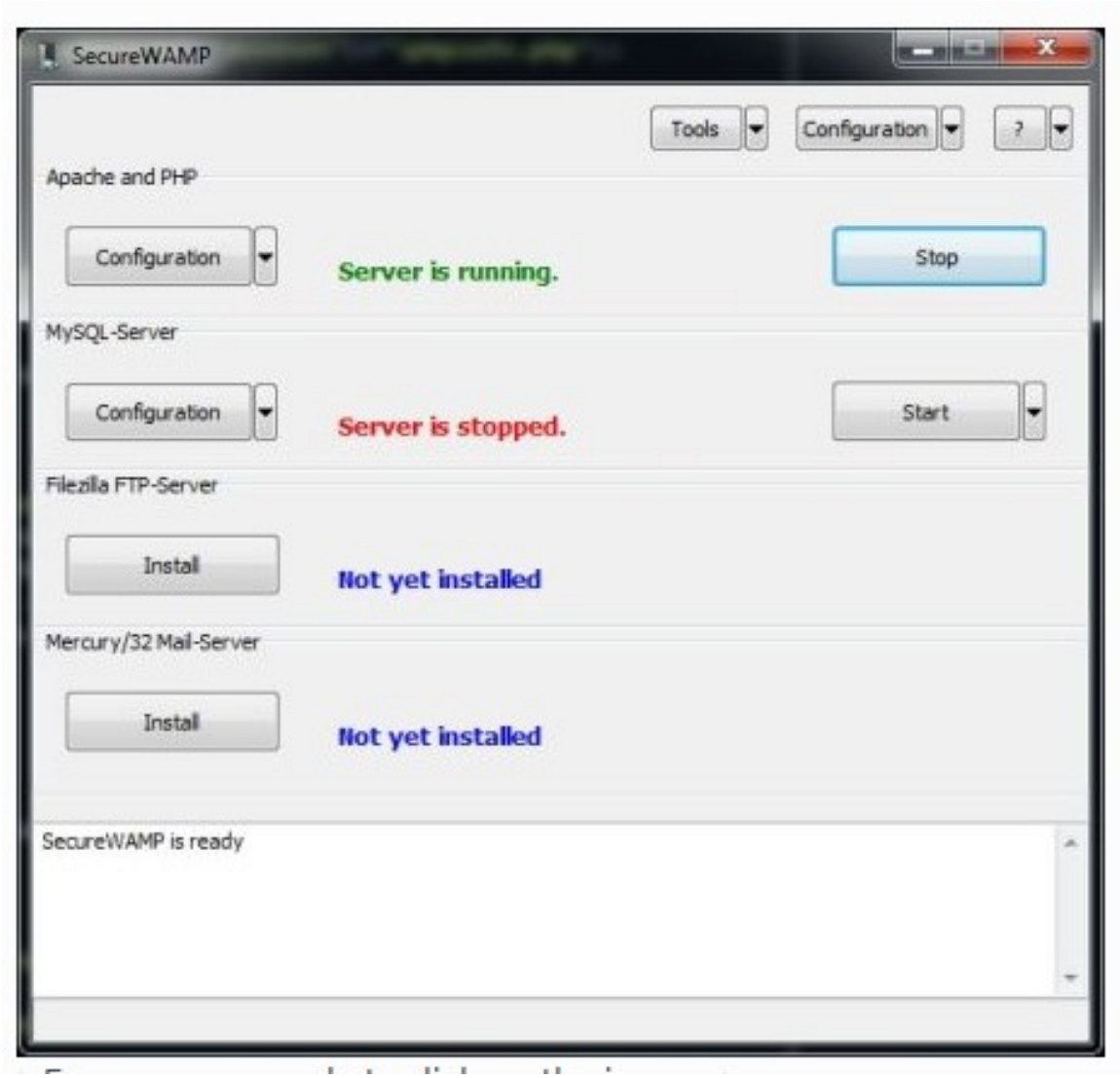
- To install My SQL workbench
- Click on the link to download https://www.mysql.de/products/workbench/ and install in the system.
- Set user name and password for the database.
- Create a database and subsequently create customer, product, vendor, material master data tables for US and Germany and insert the values into the tables.

5.3 Download the Benerator Tool

Download from the link below. We have used the latest version of benerator 0.9.8. $http://bergmann-it.de/download/download-benerator$

5.3.1 Unzipping Benerator

Unzip the downloaded file in an appropriate directory

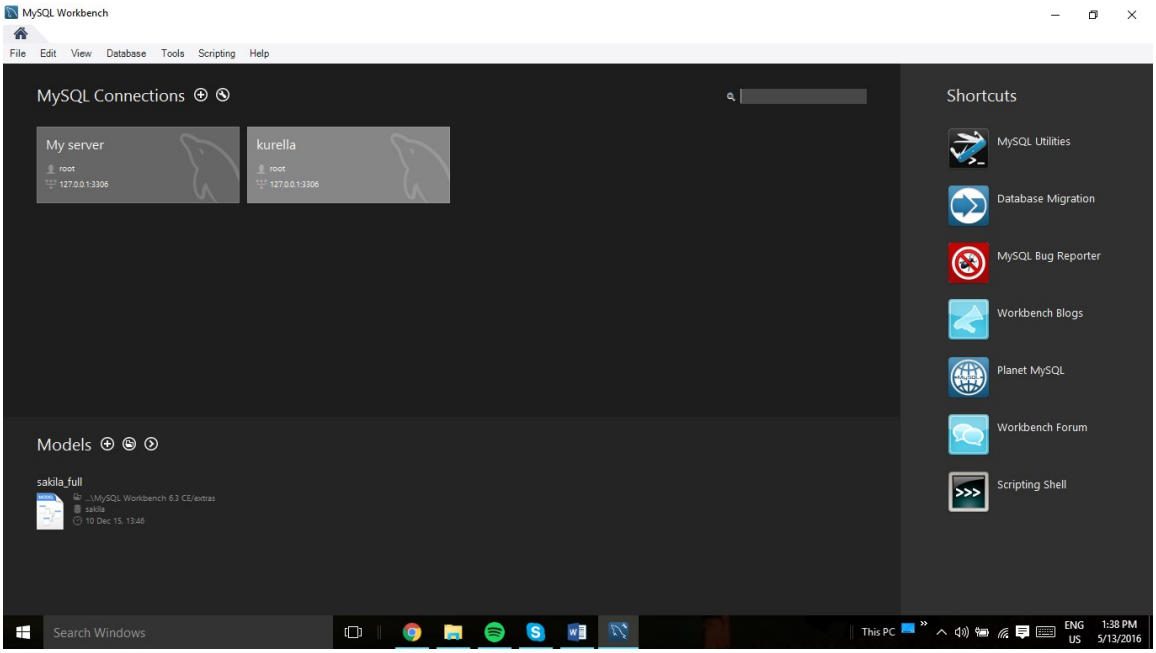


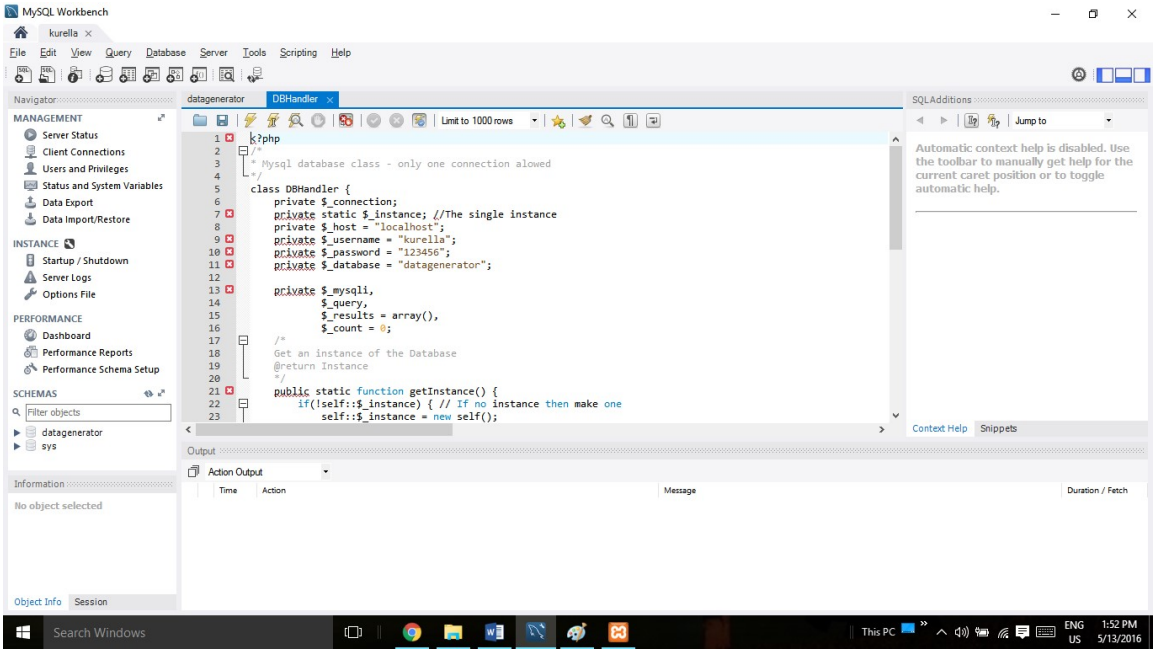
5.3.2 Set BeneratorHome

To Set in Windows system: Go to control panel, then select advanced $settings - environment$ variables. Select new in the user variable section.
Enter Benerator home as variable name and path.
IN MAC: Set $Java - home$: Enter $java - home$ as variable name and path.

5.4 Maven plugin

Download maven from $http://maven.apache.org/download.cgi$
Extract the archive, to the directory you wish to install Maven 3.3.3.
The subdirectory apache-maven-3.3.3 will be created from the archieve.
Set Maven environment variables.
Add $M2 - HOME$, M2, $MAVEN - OPTS$ to environment variables.
Set the environment variables using system properties.





$MAVEN - OPTS = -Xms256m - Xmx512m$

5.5 Verify the settings

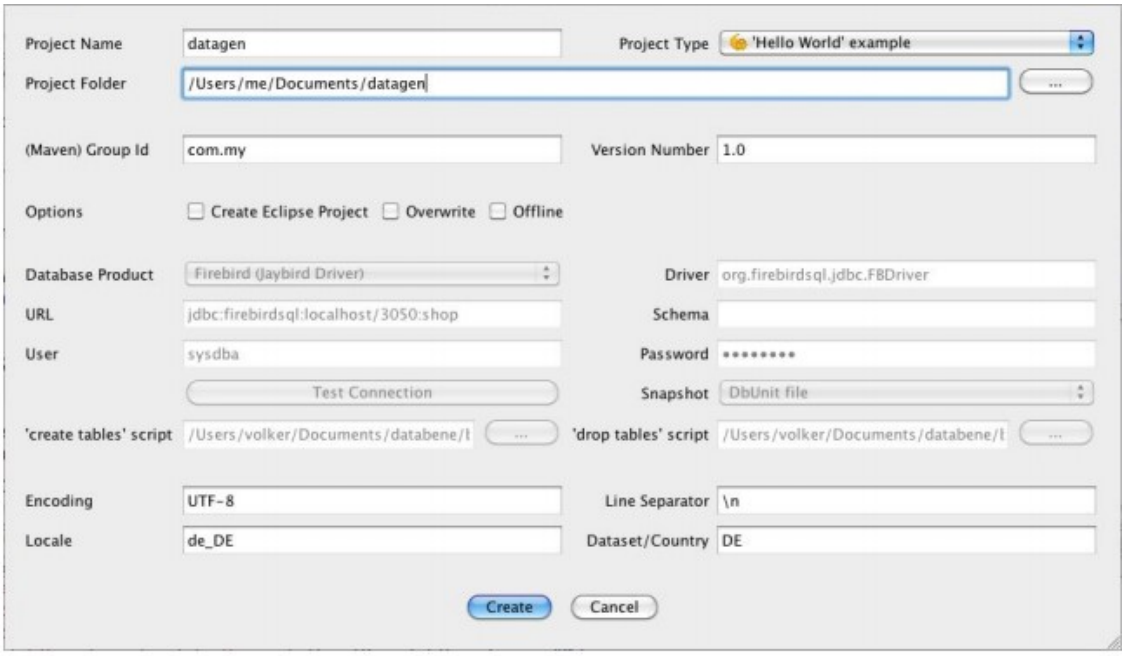
In windows, open command prompt and type *benerator – version*. In other systems like mac or Linux, type *benerator.sh – version*. Then benerator will launch and printout the version and information on java, operating systems and the script engines as shown in the screen shot.

6. Benerator Project Wizard

6.1 Start Benerator Project Wizard

Benerator Project wizard helps for easy and fast setting up of any benerator project.

- Start project wizard on command line by typing benerator-wizard-bat
- Depending on our language settings, GUI invites us in the selected language dialog.
- Then the dialog looks like as shown below



6.2 Implementation

- In the dialog box, select project type as 'populate database'.
- Create new project folder.
- Select the database product as MySQL.
- Enter the URL details: $jdbc : mysql : //localhost : 3306/ < database >$
- Enter username and password. Here we have given username as root and password as generator
- Click on create a project and then it will generate benerator.xml , pom.xml files, read me, base.dbunit and benerator in the target project folder which you have already created.
- Benerator.xml is a descriptor file which is used for the configuration purpose.
- POM.xml contains the project information and other configuration details. It also contains plugins, project dependencies, project versions etc.
- In our project,Pom.xml contains as shown below

6.2.1 Pom.xml

POM settings

- Remove the unwanted dependencies which are not related to the project.
- Change the artifact id name of 'org.databene' group id plugin from maven-benerator-plugin to benerator-maven-plugin.

6.2.2 Readme.xml

Read me file

- Read me file in the project folder contains instructions to run the commands.
- Now start database data generation by typing the command. mvn benerator:generate in the command prompt.

7. Configuration

7.1 Data Generation

```
<generate name="salesorders_us" type="salesorders_us" count="67" consumer="db,ConsoleExporter">

<id name="orderid" type="int" min="1" max="67"  />

<variable name="weightings" source="weightings01.wgt.csv" distribution="weighted"/>

 <reference name="customerid" type="int" targetType="salesorders_us" source="db" selector="select id from
customer_us" nullable="false" cyclic="true" script="{weightings}"/>

<attribute name="sale_time" type="datetime" nullable="false" generator="dtGen0901"/>

<generate name="ordersdetails_us" type="ordersdetails_us"  minCount="1" maxCount="100"
consumer="db,ConsoleExporter">

<id name="orderdetailid"  generator="new IncrementalIdGenerator"  mode="ignored" />

<reference name="orderid" script="salesorders_us.orderid"/>

<variable name="weightings01" source="Hproduct_us.wgt.csv" distribution="weighted"/>

<reference name="productid" type="int" targetType="salesorders_us" source="db" selector="select pid from
product_us" nullable="false" cyclic="true" script="{weightings01}"/>

<attribute name="count" type="int" min="1" max="20" />

 </generate>

</generate>
```

Figure 7.1: Generating entities

8. Descriptor File Format

8.1 Descriptor file

We compose the location as GB to specify Benerator to use the set of files regionalized for Great Britain. For generating data as per requirement, we import predefined data from database and CSV file by defining their respective importing classes. A populating database is to define a database that will be referred by the 'id' later a url is used to link datagenerator and mysql These entities are to be generated are configured as $Salesorder-de$ and $Orderdetails-de$.The values to be generated are defined as per requirement. The entities which are generated is kept in a variable with name weightings. The productid acts a primary key and foreign key for both entities from source file db. All attributes are declared as part of these two entities and given the name as a count has integer type having minimum one and maximum 20 values. All attributes are in the part of this sub generator are generated randomly using distribution function.

8.1.1 Setup

Benerator Configuration file is based on the XML. An XML schema is provided. The document root is a setup element.benerator descriptor files should be named as 'benerator.xml' .

8.1.2 Benerator properties

8.1.3 Comment

This prints out the output to the logger, but not to the console. We can choose an option whether to send the output or to ignore it.
<comment>generating 10000 entries for customers</comment> Commenting out a statement in between this '<!–>' denotes, we are ignoring that comment.

8.1.4 Database Repositories

For a common use databases it is helpful to use a central database configuration repositories. It can be found in the user home directory. We can define a database configuration with a name by storing in a specific named properties file. In file we can configure the JDBC connection information with

name	description	default setting
defaultEncoding	the default file encoding to use for reading and writing text files	the system's file encoding
defaultLineSeparator	the line separator to use by default	the system's line separator
defaultTimeZone	The time zone to use	The system's time zone
defaultLocale	The locale to use if none has been specified explicitly	The system's language code, e.g. 'de'
defaultDataset	The dataset to use if none has been specified explicitly	The system's country's two-letter ISO code, e.g. 'US'
defaultPageSize	the number of entities to create in one 'run', typically a transaction	1
defaultScript	The default script engine to use for evaluating script expressions	ben (DatabeneScript)
defaultNull	tells if nullable attribute should always be generated as null by default	true
defaultSeparator	the default column separator to use for csv files	,
defaultErrorHandler	the default error handling mechanism to use	fatal
validate	Boolean flag to turn off validation (e.g. of XML validity and type definition consistency).	true

$db-url$, $db-driver$, $db-user$, and $db-password$.

url ="jdbc:mysql://localhost:3306/datagenerator"

Driver ="com.mysql.jdbc.Driver"
User = root
Password = pramod02
Then we can connect to a database using the database environment attribute.
<database id>="db"

If we define a properties file in the directory in which benerator executes, this file will be used. Otherwise, configuration is taken from our database repository.

8.1.5 Schema

Schema is used as a descriptor file. It describes the various classes within a system. Benerator can generate data from a schema file, but by inserting a XML schema annotations, we can build a test data generation, which can be used for many other purposes.

schema="datagenerator"

8.1.6 Catalog

Catalog is a directory consisting of information on data sets, database or files. It describes the location where files or datasets or database entities are available and also consists of information like dataset or file are stored in which type of devices.

catalog="datagenerator"

8.1.7 Execute

It executes the code from the file using Execute.

<execute target="db" >
$droptablesalesorders_us$;
$createtablesalesorders_us($
$orderidint$,
$customeridint$,
$productidint$,
$quantityint$,
$sale_timevarchar(100)$
);
</execute>

8.1.8 Count

Depending on customer's prerequisite, count is given. For example, if we give 10,000 count, it produces 10,000 customer rows in a data.

<Comment> generating 10000 entries for customers</comment>

8.1.9 Consumer

Consumer is an object which collects the data after creating, altering and validating. It can be a storage systems, files custom JavaBeans creating consumer interface. It should not show the changes in the generated data.
$< generatetype = "sales_in_US" count = "10000" consumer = "db, ConsoleExporter" >$

8.1.10 Bean Id

Classes in Benerator which provide existing functionality. $< beanid = \blacksquare dtGen \blacksquare class = \blacksquare DateTimeGenerator \blacksquare / >$

8.1.11 Variable

For a complex data generation we can use the variables in the descriptor files. They are placed inside an element as
$< variablename = "person" generator = "PersonGenerator" / >$

8.1.12 Property Name

The enclose property statements will make JavaBean properties and attributes for setting an appropriate values.

8.1.13 Attribute

A single data unit or item relating to a data object. It is otherwise called as column or field also.

$< attributename = "quantity" type = "integer" min = "0" max = "10" distribution = "cumulated" / >< attributename = "sale_time" type = "datetime" generator = "dtGen" / >$

8.1.14 Import

Benerator will provide an import option similar to a java language. We can import domains, classes and packages. So instead of using the qualified name like bean id as
$< beanid = "dtGen" class = "DateTimeGenerator" >$ We can use import class and use a local name as $< importdomains = "person" / >$

8.1.15 Generate

$< generatetype = "salesorders_us" count = "10000" consumer = "db, ConsoleExporter" >$
$< idname = "orderid" type = "integer" / >$
$< referencename = "customerid" type = "int" targetType = "salesorders_us" source = "db" selector = "selectid fromcustomer_us" nullable = "false" cyclic = "true" distribution = "random" / >$
$< referencename = "productid" type = "int" targetType = "salesorders_us" source = "db" selector = "selectpid fromproduct_us" nullable = "false" cyclic = "true" distribution = "random" / >$
$< attributename = "quantity" type = "integer" min = "0" max = "10" distribution = "cumulated" / >$
$< attributename = "sale_time" type = "datetime" generator = "dtGen" / >$
$< /generate >$

By using the above descriptor file formats, we have generated the data as shown in the following chapter.

8.2 Weight Distribution Function

Weight distribution function In order to generate the realistic datasets for analyses purpose, we have used math tool weight function. With this principle, we can control the input parameters in the Benerator.xml file along with the automatic generation of datasets.
Weight function determines the probability of actual value and allows you to gives a weight. While using this function, Benerator will generate datasets in random order for all applicable numbers in order to make normalize according to our preferences. There is no need to use normal distribution function for every data set. When importing original data from a CSV file, every weight is expected to be in an additional row. If one row has more than one column then the capacity of the second column is made clear as weight. A weight is assumed as one when there is no column and Benerator normalizes automatically for all data objects. It is explained by a class that implements the interface You can weigh any numeric data. You can give different weights to the customers using weight function and this data stored in a comma separated value (CSV) file using a suffix

like '.wgt.csv' and the filename given as 'weightings' . $<$ *variablename* $=$ "*weightings*"*source* $=$ "*weightings.wgt.csv*"*distribution* $=$ "*weighted*"/ $>$

Products	weight percentages
1	1.9
2	2.5
3	3.3
4	2.8
5	4.2
6	3.9
7	4.1
8	1.7

9. Data Generation

- Now we have configured the benerator.xml descriptor file as per our requirements.
- Set the path of the project folder in the command line.
- Run the command mvn benerator:generate in command prompt to generate the data.

orderdetailid	productid	orderid	count	orderid	customeri	sale_time	pid	pname
1	15	1	17	1	1	22-01-2009	15	PROFESSIONAL TOURING BIKE (BLACK)
2	11	1	6	1	1	22-01-2009	11	DELUXE TOURING BIKE (BLACK)
3	6	1	20	1	1	22-01-2009	6	REPAIR KIT
4	9	1	20	1	1	22-01-2009	9	WATER BOTTLE
5	12	1	5	1	1	22-01-2009	12	DELUXE TOURING BIKE (RED)
6	18	1	12	1	1	22-01-2009	18	WOMEN'S OFF ROAD BIKE EN
7	13	1	20	1	1	22-01-2009	13	DELUXE TOURING BIKE (SILVER)
8	11	1	1	1	1	22-01-2009	11	DELUXE TOURING BIKE (BLACK)
9	11	2	20	2	4	31-01-2009	11	DELUXE TOURING BIKE (BLACK)
10	3	2	6	2	4	31-01-2009	3	FIRST AID KIT
11	5	2	1	2	4	31-01-2009	5	OFF ROAD HELMET
12	14	2	4	2	4	31-01-2009	14	MEN'S OFF ROAD BIKE
13	9	2	18	2	4	31-01-2009	9	WATER BOTTLE
14	16	2	10	2	4	31-01-2009	16	PROFESSIONAL TOURING BIKE (RED)
15	3	2	2	2	4	31-01-2009	3	FIRST AID KIT
16	12	2	17	2	4	31-01-2009	12	DELUXE TOURING BIKE (RED)

Figure 9.1: Sales in Germany from 2009-2011

similarly we can create a sales data for US from 2009-2011.

code	pnum	type	price	id	customerid	postalcode	city	customername	countrycod
EN	PRTR1000	FG	510	1	13000	60549	FRANKFUI	AIRPORT BIKES	DE00
EN	DXTR1000	FG	380	1	13000	60549	FRANKFUI	AIRPORT BIKES	DE00
EN	RKIT1000	TG	45	1	13000	60549	FRANKFUI	AIRPORT BIKES	DE00
EN	BOTL1000	TG	5	1	13000	60549	FRANKFUI	AIRPORT BIKES	DE00
EN	DXTR3000	FG	400	1	13000	60549	FRANKFUI	AIRPORT BIKES	DE00
EN	ORWN1000	FG	320	1	13000	60549	FRANKFUI	AIRPORT BIKES	DE00
EN	DXTR2000	FG	410	1	13000	60549	FRANKFUI	AIRPORT BIKES	DE00
EN	DXTR1000	FG	380	1	13000	60549	FRANKFUI	AIRPORT BIKES	DE00
EN	DXTR1000	FG	380	4	16000	16341	BERLIN	CAPITAL BIKES	DE00
EN	FAID1000	TG	50	4	16000	16341	BERLIN	CAPITAL BIKES	DE00
EN	OHMT1000	TG	20	4	16000	16341	BERLIN	CAPITAL BIKES	DE00
EN	ORMN1000	FG	310	4	16000	16341	BERLIN	CAPITAL BIKES	DE00
EN	BOTL1000	TG	5	4	16000	16341	BERLIN	CAPITAL BIKES	DE00
EN	PRTR3000	FG	510	4	16000	16341	BERLIN	CAPITAL BIKES	DE00
EN	FAID1000	TG	50	4	16000	16341	BERLIN	CAPITAL BIKES	DE00

Figure 9.2: *Sales – in – Germany* from 2009-2011

10. Analyses and Results

10.1 Region-wise customer distribution

We have created the sales by taking 4 reasons into consideration according to city-wise distribution.city topology, Population, Bike users and City infrastructure of lines. There are 12 customers for Germany, from 11 cities, and 13 customers for the USA from 13 cities. in Germany,we can see from the following figure that sales varies in each city based on the considered reasons.Hamburg has the highest number of sales because of two customers.Similarly region wise sales for US also generated

10.2 Seasonal-wise customer distribution

Seasonal-wise distribution: In the real world, sales never remain same. It changes every day or month. it may be due to climatic conditions or geographical changes. Generally bike sales will be high during summer time (May-October) because of pleasant weather for the bicycle riders to go out. But during winter,due to heavy snowfall or rains or other climatic issues riders prefer other transport rather than bicycles. According to our generated data from 2009-2011,we have seen that Sales in both Germany and the USA have increased in summer and decreased in winter. Particularly when we have considered the sales per every year,Germany sales have increased in 2009, 2010 and 2011. Where as, sales in the USA have increased in 2009,2010 but decreased in 2011 as shown in figure.

10.3 Product preference

Generally, every customer before buying a product, he has some preferences.Based on there preferences customer decides to buy a product. In this generated sales,customer has bought a product based on two interesting things.

10.3.1 Colour

Colour: From both touring and off-road bikes with three different colours black ,silver and red, We found that many of the customers showing interest to buy black coloured bikes. Silver coloured bike

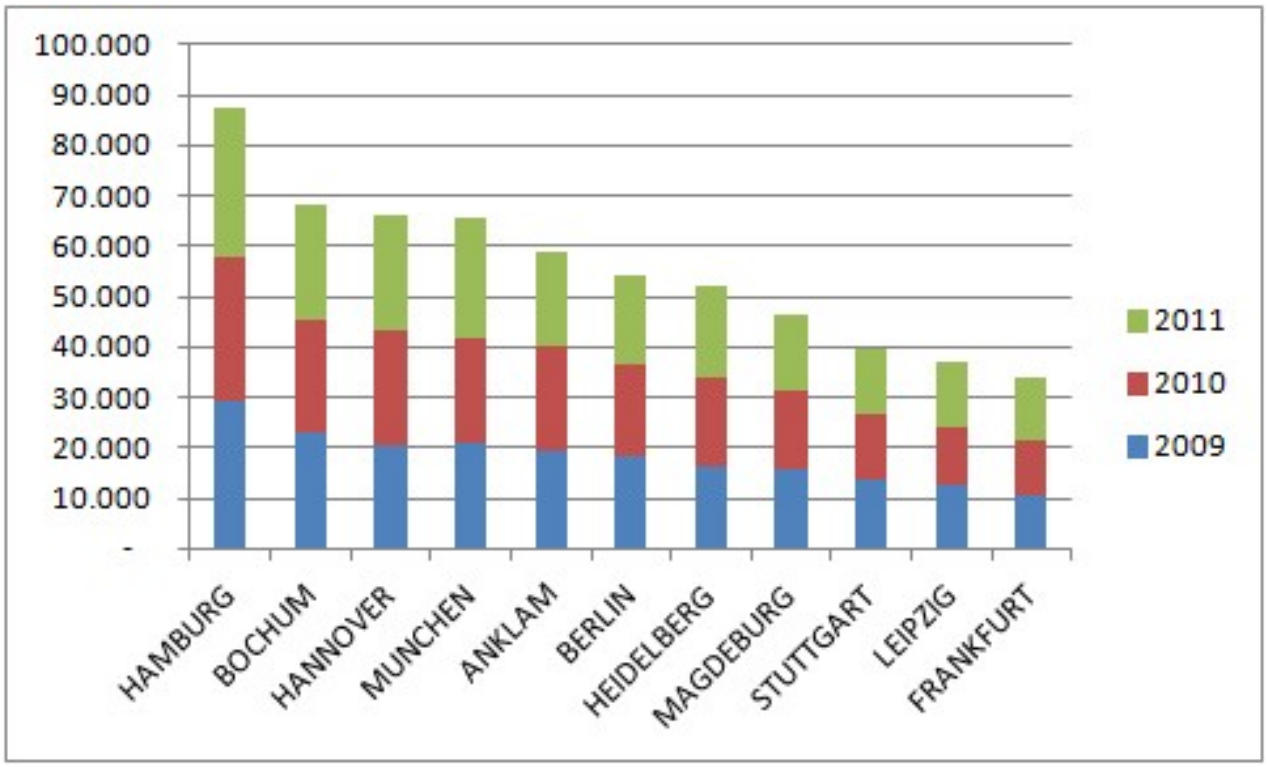


Figure 10.1: Region-wise in Germany from 2009-2011

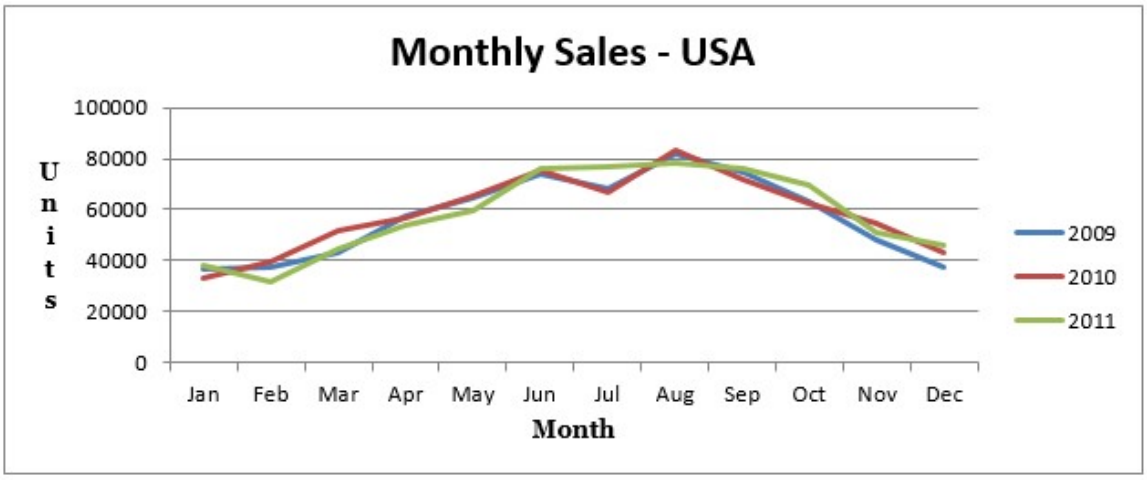


Figure 10.2: yearly sales in US from 2009-2011

sales were in second position with least preference to red colour. This sales were shown in figure

10.3.2 Accessories

Along with the bikes, GBI also sells Accessories like Helmet,First-aid kit,water bottle etc.When compared to product sales, accessories sales were low as shown in figure for both Germany and US for three years 2009-2011. We also found that more number of men prefered to ride bike when compared to women. Therefore, with these analyses, we found that our Benerator tool is meeting all

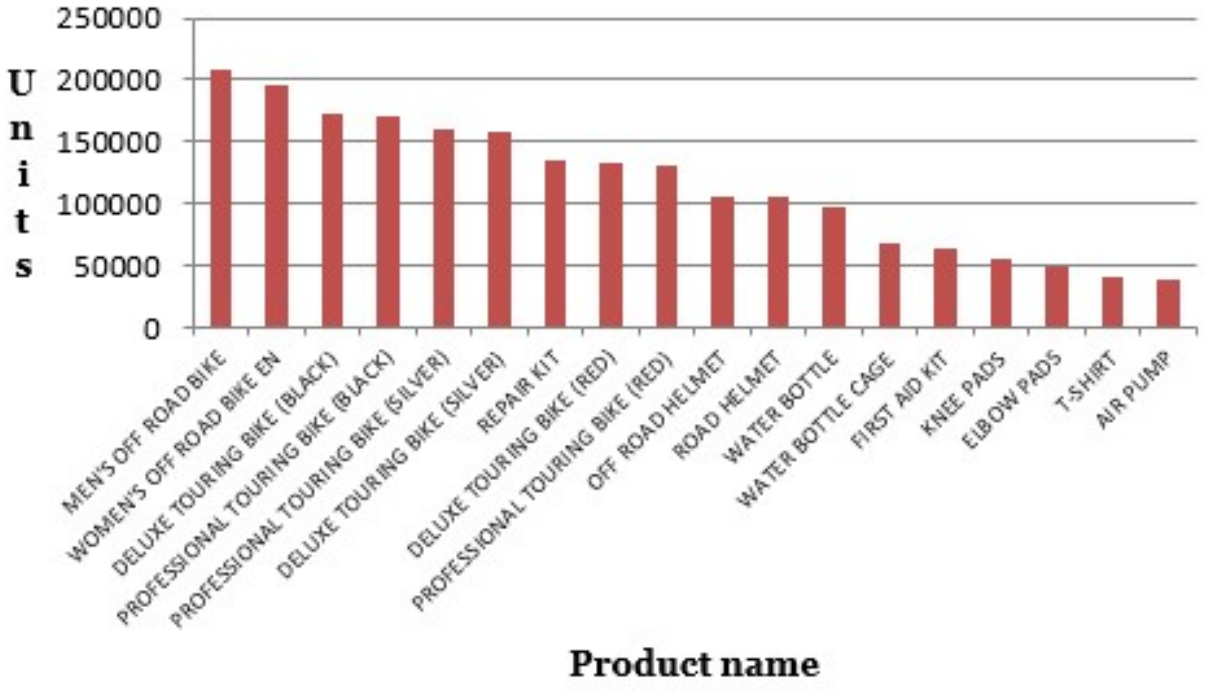


Figure 10.3: Products and Accessories

the neccessary requirements to generate a realistic synthetic data which is approximately equal to the real data.

11. Conclusion

Finally, as per the requirements and data types for the entities given by user in the Benerator.xml which is a descriptor file, Benerator has generated the realistic synthetic data in the required output format as per the selection. This output data is from the database is exported into Excel sheet.MYSQL database is also employed in the implementation of data generation to store the attributes of generated data. various analyses has also done to show that our generated realistic data is approximately same as real data.

12. Appendix

12.0.1 Learnings and Performance

- Benerator can be used to generate realistic datasets.
- Documentation about the functionality of the tool is not clearly explained. Even the forum is not active from past 5 years for further suggestions.
- To Generate 600,000 rows of realistic datasets for Germany, It took nearly 15 hours.So,performance is low.

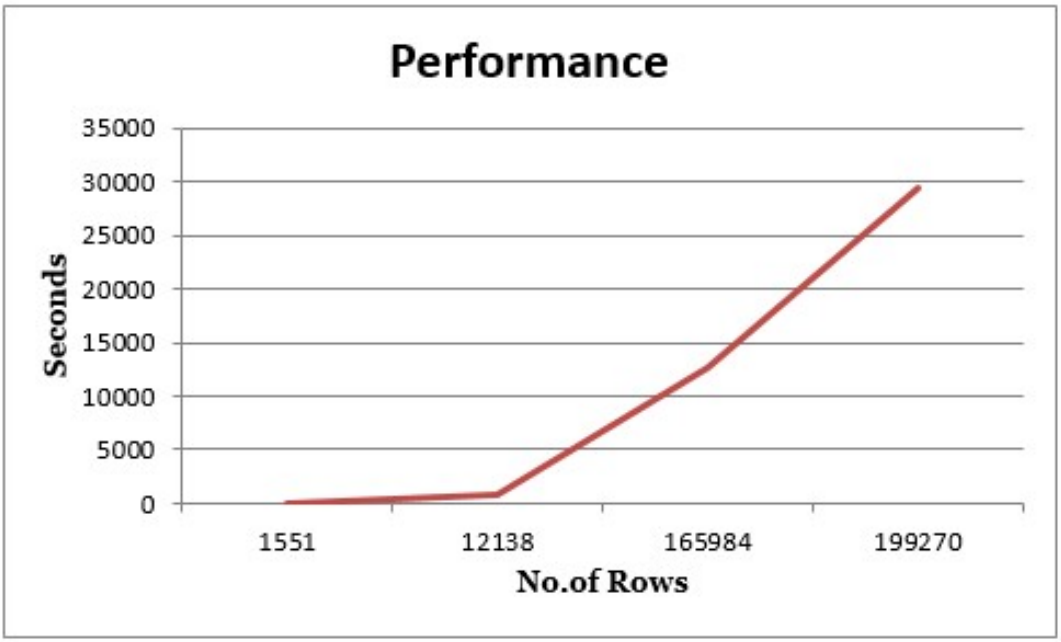


Figure 12.1: performance of Benerator Tool

13. Acknowledgement

The authors would like to Thank SAP UCC Magdeburg and Otto-VOn-Guericke University Magdeburg for their guidance in successful completion of the Paper. Further, this project is successfully finished with the effort and coordination of our Team members.

13.1 References

** Wish you all the best **