# Proposal for Standardising Data - International / Indian Standard Organisation Number


Proposer Name : Vinod Krishnan Menon

E-Mail ID : vk196808@gmail.com

Contact No. : 9481724415

Address : 351, 1st Floor, 2nd Cross,
1st Block,
R.T. Nagar,
Bangalore - 560032,
Karnataka,
India.


Date : 12/09/2020

| | | |
|---|---|---|
| 1 | Proposer Name and Contact Details | Proposer Name : Vinod Krishnan Menon<br>Address : 351, 1st Floor, 2nd Cross, 1st Block<br>R.T. Nagar,<br>Bangalore - 560032,<br>Karnataka,<br>India.<br>E-Mail ID : vk196808@gmail.com<br>Contact No. : 9481724415 |
| 2 | Proposed Title Of Standard | International/Indian Standard Organisation Number (ISON) |
| 3 | Scope Of Proposed Standards | Standardisation of Master Data – Customer Master/ Vendor Master Numbers and other such numbers and codes used for organisations. |
| 4 | Purpose And Justification | Please refer the discussion titled “Purpose and Justification” at the end of this table. |
| 5 | Likely Users Of Standards : | All Organisations, Information Technology Companies and every organisation which uses information technology to keep track of the organisations that they interact with such as customers, vendors etc and their related data. All those who |

| | | |
|---|---|---|
| | | are interested in using Big Data to perform various data analysis. E-Commerce Organisations. |
| 6 | Any Related Standards/Series of Standards/System Standard required to make this Subject Standard complete | Not Applicable |
| 7 | When the final Standard would be required (any time limit) | No time limit. |
| 8 | Any specific problem being faced without this Standard | Please refer the discussion titled “Purpose and Justification” at the end of this table as well as the section on Benefits of standardisation. |
| 9 | Bearing with Government legislation requlation etc. | Not Applicable |
| 10 | Name and address of manufacturers/implementing/ industries/purchasing organisation/component supplier/raw material supplier , if any. | Not Applicable |
| 11 | Status of the industry in the country. | Currently propreitary data is being used instead of standardised data. |
| 12 | Availablity of test facilities in the country. | Not Applicable |
| 13 | Whether related to variety reduction, export,health | Not Applicable |
| 14 | Relevant Supportive Document | SQL scripts used to create test database are appended to the end of this document. |
| 15 | R&D Work done in India | SQL scripts used to create test database to demonstrate the problem. |
| 16 | Any foreign Collaboration | Not Applicable |
| 17 | Liason with any organisation(s) | Not Applicable |
| 18 | Preparatory Work | SQL scripts used to create test database to demonstrate the problem. |
| 19 | Whether this project can be funded by your organisation | Not Applicable |
| 20 | Whether your organisation would be interested to opt for BIS Standard Mark once the standard is published. | Not Applicable |
| | | |

**Purpose and Justification**

This is a proposal for the standardisation of master data which exist in the master tables of computer databases used by all types of organisations, especially business organisations. This proposal in particular deals with the creation of an International/Indian Standard Organisation Number which can be assigned to all organisations. The International/Indian Standard Organisation Number is intended to replace the Customer Number, Vendor Number and any other proprietary numbers or codes which may be assigned to an organisation by the database owner or even by third parties. The International/Indian Standard Organisation Number should be applicable not only for all business organisations but for all types of organisations including educational, medical, charity, retail organisations and any other type of organisation including perhaps kirana stores and even agricultural farms.

In the course of my career in the Information Technology field, I have seen many databases most of which contain a vast number of tables which store a wide variety of data. These tables could be classified into two types namely master tables and transactions tables. Two of the common master tables which are found in the databases of business organisations are the Customer_Master and the Vendor_Master tables. These tables are basically a list of customers and vendors that the organisations have dealt with in the past or do so in the present. There would be one record in the Customer_Master for each customer and likewise there would be one record in the Vendor_Master for each vendor. The Customer_Master and Vendor_Master are at their very core a list of addresses of all the customers and vendors of the organisation. There may be other data contained in seperate fields in these tables which may contain other information about the vendors and customers that the organisation may need to keep track of. The structures of these tables can vary from organisation to organisation but would have in common a set of fields containing the address of these organisations.

Additionally each of the customers and vendors would be assigned a unique number in these tables which would be called the Customer Number in the Customer_Master and Vendor Number in the Vendor_Master. These numbers are unique numbers which are assigned to each customer and vendor by the organisation using a proprietary system which may be convenient for that organisation. In other words these numbers are proprietary numbers and are specific to that organisation.

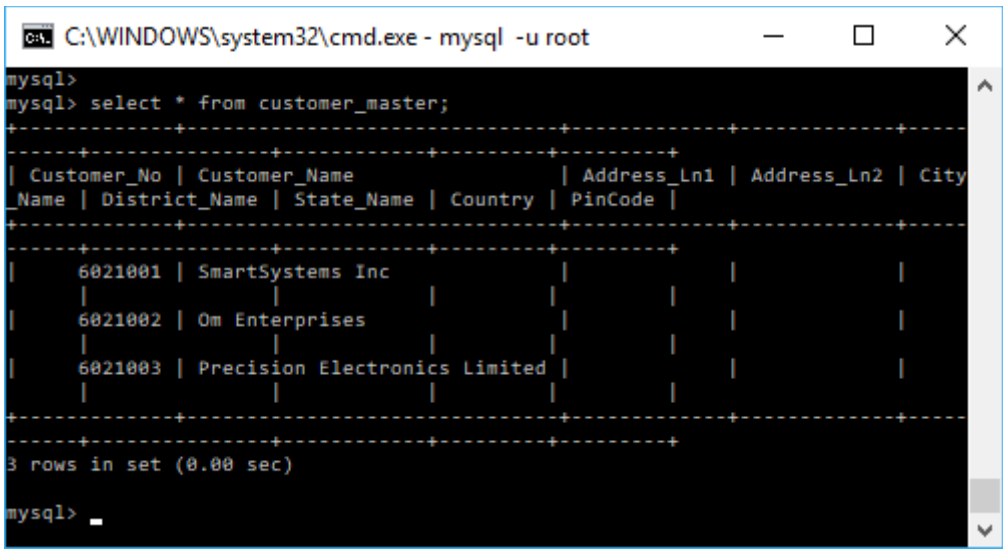
In some cases, the customer of the organisation could also be a vendor. In such as situation there would be an entry for that organisation in the Vendor_Master as well as in the

Customer_Master tables. Since the customer numbers and vendor numbers are proprietary numbers and the numbering sequence for both could be quite different, it is possible that the same organisation could be assigned one Customer Number and an entirely different number as the Vendor Number. The number of such cases where the customer of the organisation is also a vendor can vary from organisation to organisation. In some organisations, a few, perhaps a small percentage, of the customers could also be vendors. In others, for example, 50% of the customers could also be vendors and perhaps there could be some cases where all of the customers are also vendors. In some organisations, it would be necessary to keep track of a given customer's corresponding Vendor Number, so an additional field is added to the Customer_Master to store the appropriate Vendor Number and in the Vendor_Master likewise there could be a field to store the appropriate Customer Number for that organisation.

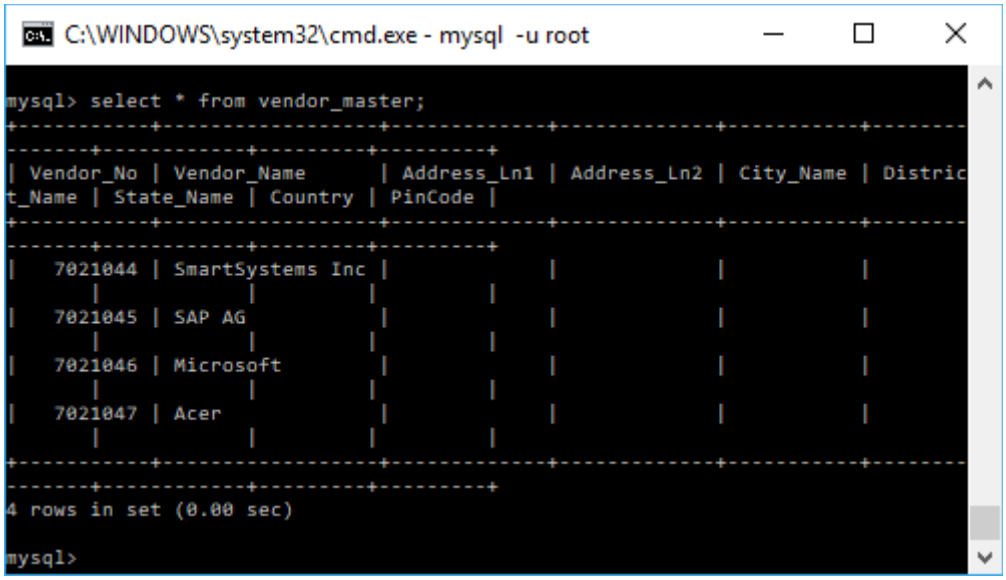
The situation in the databases maintained by many organisations can be demonstrated using data from a test database containing a Customer_Master and Vendor_Master tables which contain a few records only as sample data. These tables could have many fields other than what is shown in the screen shots below, however only a few basic fields as may be required have been shown for this example.

A MySQL test database was created and a Customer_Master table and Vendor_Master table were also created within the database having the respective Customer and Vendor Number fields and the address fields only. These two tables were populated with test data records. Only the fields Customer_Number, Customer_Name, Vendor_Number and Vendor_Name have been populated with data and the rest of the fields are left empty as this is sufficient for the current purpose. Then various SELECT statements which provided results as described below were executed on these tables.

The screen shot below shows the results of a SELECT statement executed on the Customer_Master table where all records and all fields in the table are selected.




Now the same SELECT statement is executed on the Vendor_Master table and all fields and all records which exist in the table are displayed in the screen shot below.



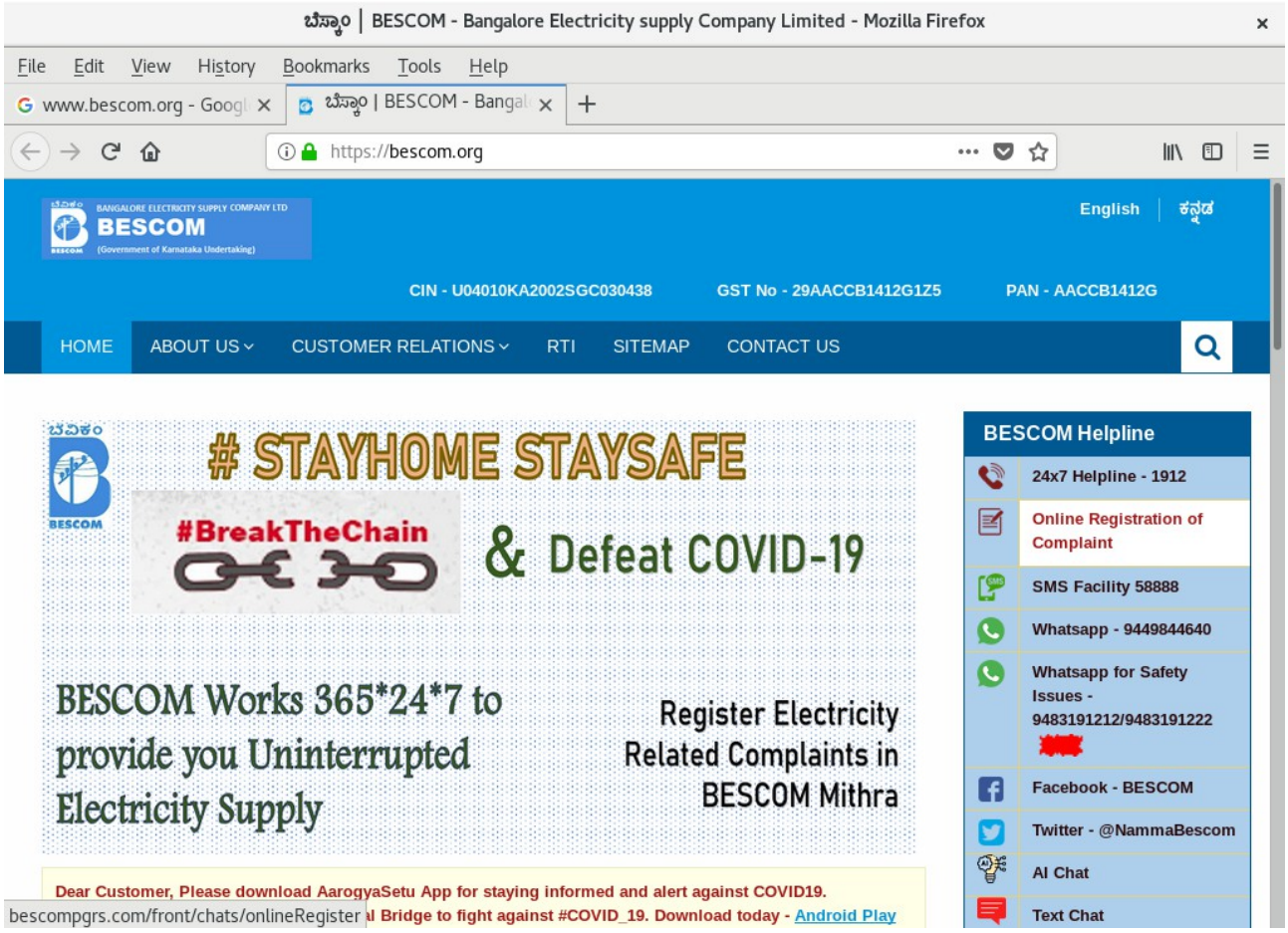

In the screen shots above it can be seen that the Customer Number 621001 and the Vendor Number 7021044 refer to the same organisation. These two numbers are proprietary Customer_Numbers and Vendor_Numbers which have been allocated to this organisation by the proprietary alogrithm used by the database owner. Further more, this organisation with the Vendor_Number 7021044 would do business with many other organisations and would have a customer and or vendor relationship with these other organisations. The Customer_Number and Vendor_Number that this organisation may have in the databases of these other organisation could be completely different again. So here we would have situation where an organisation could be known by many different Customer_Numbers and Vendor_Numbers across the databases of all the

organisations that it may interact with.

So the thought flashed through my mind, why should an organisation be known by so many proprietary numbers in the databases of the organisations it interacts with. In the science of biology, each species is given a unique name consisting of genus and species to uniquely identify it. In any database or research paper, if the scientific name of a creature is specified, it would be clear to who ever is reading it as to what particular creature is being refered to. Likewise in the case of Information Technology/Computer Science, why not create a Standard Organisation Number which could replace all these proprietary Customer and Vendor numbers and other relationship numbers in the many databases.

Every production database in an organisation records all the transactions done by that organisation with all other organisations it may interact with in one way or the other. The sum of all these databases in the country put together would be a record of all the transactions that have been done in the country. If all databases were to use the same number or code to refer to the same organisation or object or entity then many possiblities arise.

Another example of proprietary numbers assigned to a particular organisation is the case of many third parties assigning their own respective proprietary numbers or codes to other organisations. An example of this can be shown by using the case of BESCOM. The screen shot below displays the opening page of the website belonging to BESCOM. At the top of the page three numbers can be seen, namely CIN, GST No and PAN. These three numbers are proprietary numbers assigned to BESCOM by three different organisations. BESCOM is known to these three organisations by their respective unique or proprietary numbers or codes.




Many questions now arise. Why is it necessary for three different organisations to use three different unique numbers or codes to identify the same entity namely BESCOM ? Why is it not possible for a Standards Organisation to assign a Standard Organisation Number for each Organisation ? Why is it not possible for these three organisations to use the same Standard Organisation Number to identify BESCOM in their databases instead of assigning their own proprietary numbers or codes to BESCOM ? Why not replace the CIN, GST Number, PAN Number and any other such number or code with a Standard Organisation Number ?

A similar situation exists in the case of each citizen in India. Every citizen is allocated a PAN Number by the Income Tax Department. Every citizen is allocated a Voter Id Card with unique number by the Electon Commission. Each citizen is assigned an Aadhar Number and Card by the UIDAI. Every Bank would give their Customers a unique Customer Number which is propietary to that Bank. The PAN number is the number by which the Income Tax Department knows the citizen by. The Voter id is the number by which the Election Commission knows the individual. Likewise each organisation that the individual interacts with would identify the individual using the proprietary Customer Number that they have assigned to that individual. Therefore the question arises why should an individual be identified by so many different numbers, each proprietary to the

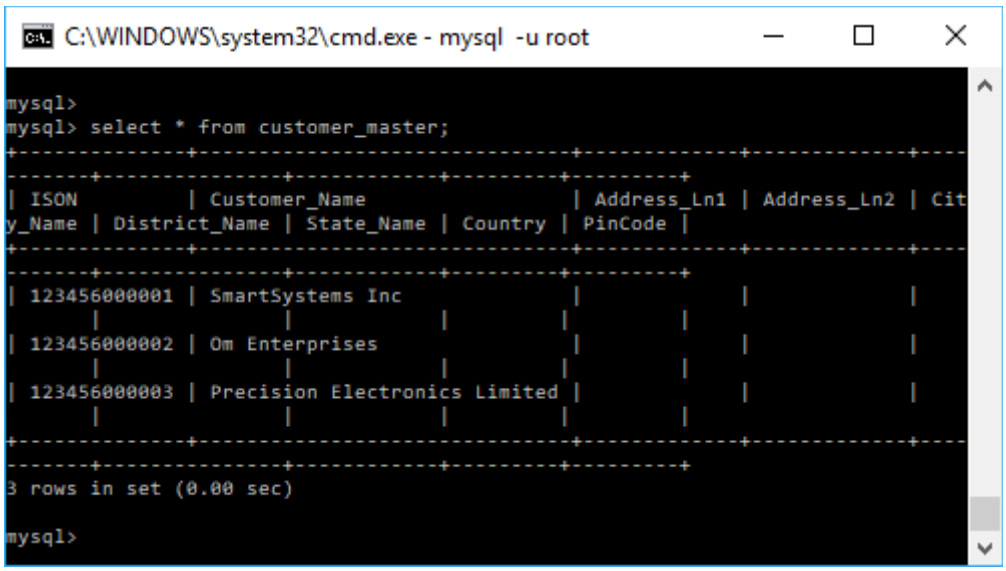
organisation which the individual interacts with. A single standard number could be assigned to the individual which could replace all these proprietary numbers. The Aadhar number could perhaps play this role. However this is a controversial issue and there are people who are worried about their privacy.

In the case of a standard organisation number however there is not likely to be as much controversy and it has many potential benefits which could be of use to the organisations themselves. Some of these potential benefits will be discussed under the heading "*Benefits of a Standard Organisation Number*" below.

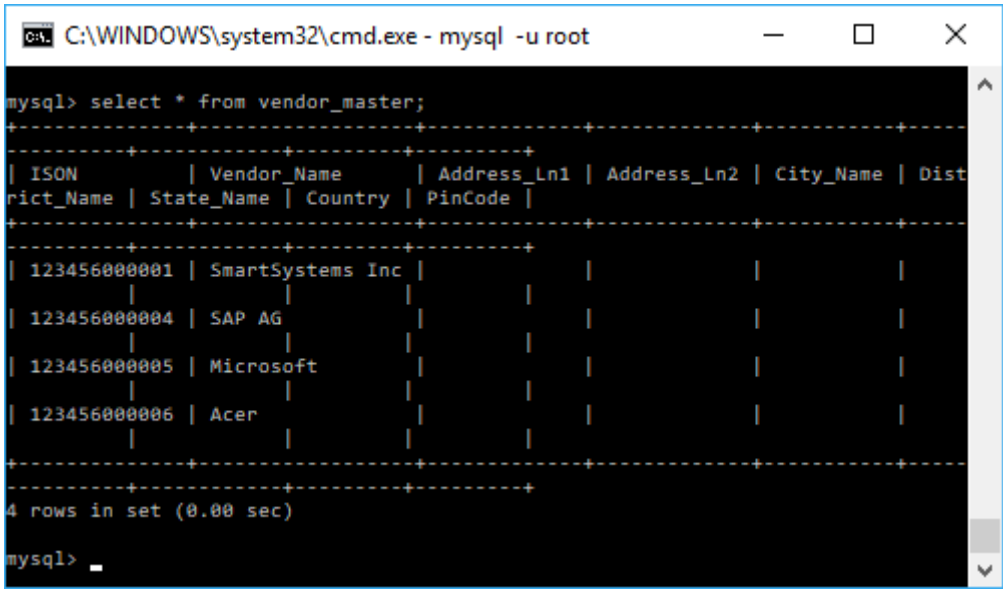
Therefore I propose that an organisation engaged in the creation of standards such as the International Standards Organisation or Bureau of Indian Standards should develop a Standard Organisation Number for all organisations. Since any organisation could potentially do business with any other organisation across the world, it would be prefereable that this standard organisation number be an international one. If for any reason it is not possible to create an International Standard Organisation Number, then the Standard Organisation Number could be created specifically for use by Indian Organisations and could be called as the Indian Standard Organisation Number. Either way, the Standard Organisation Number could be known by the acronym ISON which can stand for International/Indian Standard Organisation Number.

In the database used for the examples above, the Customer_Number and Vendor_Number in their respective tables could be replaced with a field called as ISON and the respective ISON assigned for each organisation.

The SELECT statement to extract all records and all fields from the Customer_Master table could then be executed. The results obtained would be as shown in the screen shot below.




The same SELECT statement to extract all records and all fields from the Vendor_Master can then be executed to see what the table contains. The results would be as shown in the screen shot below.

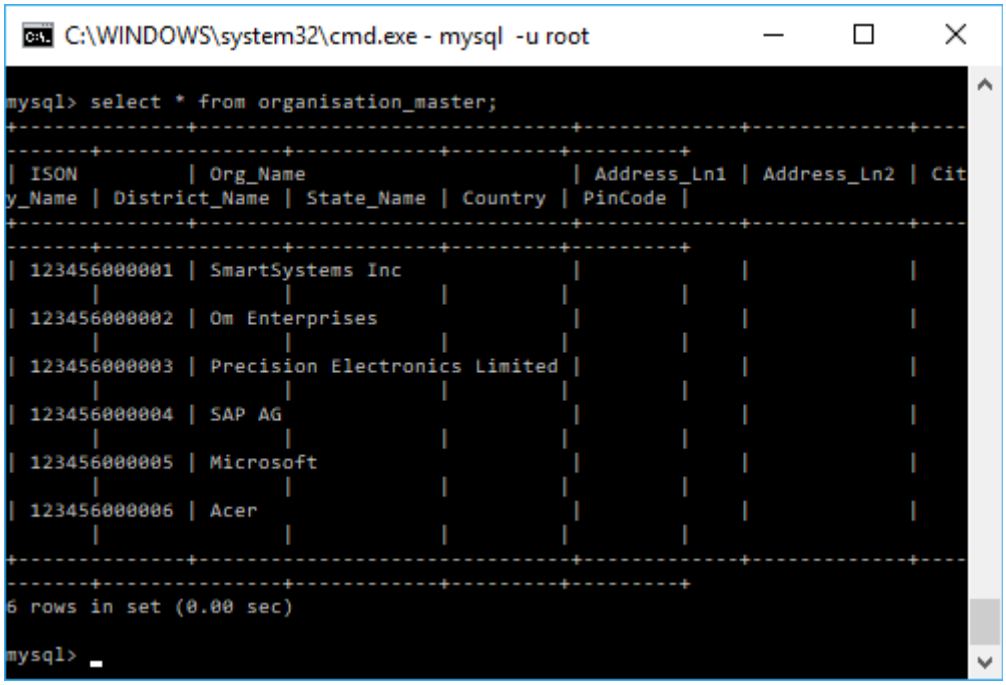



This time it can be seen that the organisation which previously had the Customer Number 621001 and the Vendor Number 7021044 now have been assigned the same Standard Organisation Number.

This gives us the possiblity of replacing the Customer_Master and Vendor_Master tables with only one table which can be called the Organisation_Master as it would contain a record for all the organisations with which the entity owning the database interacts with. The screenshot below displays the results of executing a SELECT statement on the Organisation_Master table in order to

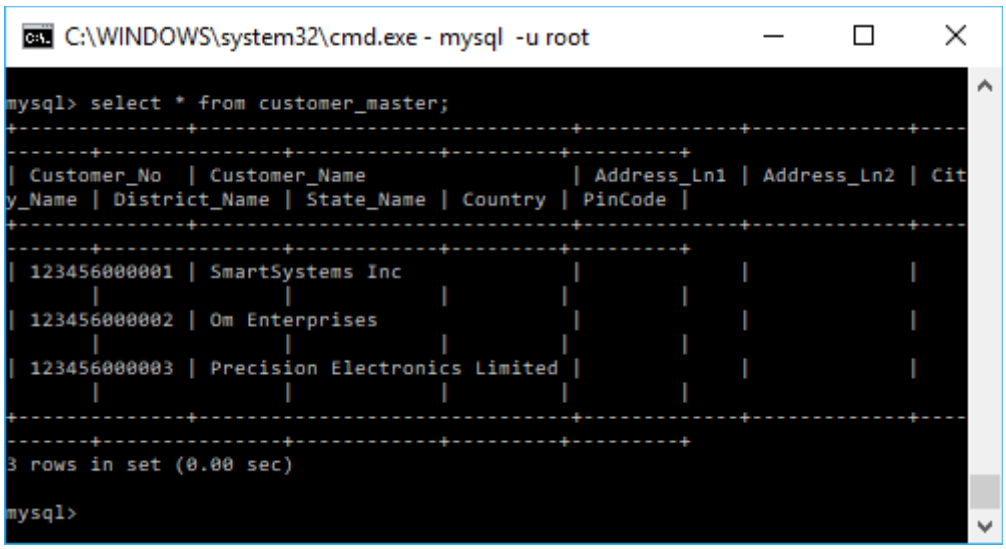
retrieve all rows and all fields of the table.

It can be seen that the resultset contains a listing of all organisations regard less of their relationship with the organisation owning the database. The relationship could be as customer or vendor or any other.

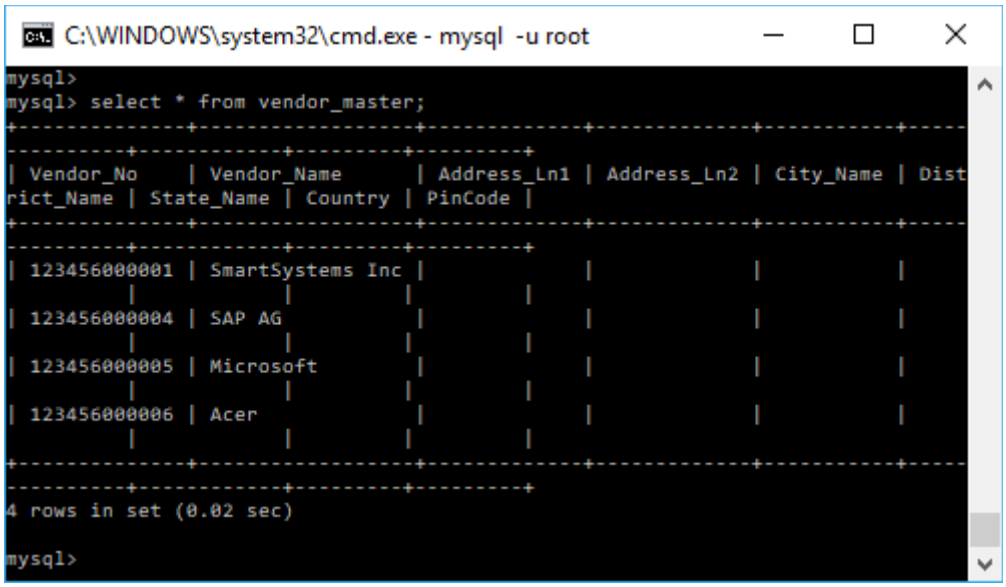



The Customer_Master and Vendor_Master can be implemented as views on the Organisation_Master table. Refer to the SQL scripts at the end of this document for details on how this is done.

Once the Customer_Master and Vendor_Master are created as views, queries can be exectued on them to find out what they contain. Executing a SELECT statement to retrieve all rows and all fields from the Customer_Master view would give the resultset shown in the screen shot below.




Executing a SELECT statement to retreive all records and all fields from the Vendor_Master view would give the resultset as shown in the screen shot below.




We find that the output of these views is the same as when these master tables were implemented as actual tables. The source of the data provided by each view is a single table. This single table contains the list of all the organisation that the database owner interacts with. This would make sure that there would be no duplicate records which are likely to exist in the Customer_Master and Vendor_Master tables in situations when a organisation is both a customer as well as a vendor.

Apart from the Customer_Master, Vendor_Master the database could have many other

master tables which contain proprietary data which could also be potentially standardised where ever practical. Over a period of time all or at least most of the proprietary data in the databases of organisations can be gradually phased out and replaced with standard data.

**Benefits of a Standard Organisation N umber**

Some of the benefits of developing an International/Indian Standard Organisation Number are listed below.

- In any economy there are two main types of entities namely organisations and individuals which interact with each other. These two entities trade in goods and services amongst themselves and between themselves. Organisations do business with other organisations. Individuals may do business with other individuals and organisations may do business with individuals. The nature of the business relationship could vary, it could be as customer, vendor, subcontractor, dealer, distributor, and so on. An organisation should be known by the same organisation number in any database regardless of the kind of relationship between the organisations. Likewise all goods and services should be identifiable using the same code or number in any database. For example, an AA battery should be known by the same number regardless of whether it is in the database of the manufacturer of the battery or in the databases of any retail outlet which sells the product to the end user. The development of such standards can be compared to the development of scientific names of every species of plant and animal so that scientists can identify species in their works without resorting to using common/local/proprietary names which could result in confusion and errors. Thus It would be logical and rational and appropriate to ensure that all organisations and individuals, goods and services be known by the same codes, numbers etc in every database in the same manner that an animal or plant is know in scientific papers, books , journals and other such literature.
- Databases of various organisations contain a lot of data accumulated over the years. The data in these databases could be used for a wide variety of analysis. All those who would like to perform such analysis on Big Data Databases would find it easier if every organisation, individual, goods and services are known by the same codes or numbers in every database. If international standard organisation numbers are created and then used widely across all the databases which exist in the world, then it would be possible to perform a search in any production database across the world for any information pertaining

to the organisation of interest. For example if an organisation were to be assigned the standard organisation number of 123456000001 then it would be possible to query any database for data pertaining to this organisation.

- When a company takes over another company, it may find it necessary to merge the databases of that company with its own databases so that there is a centeralised database to manage the combined entity. In such a situation if the databases contain standardised data, merging these databases could be comparitively easy. The Organisation_Master of one company could be merged with the Organisation_Master of the other. Since the data is standard in nature, there could be records in the source which could aleady exist in the destination master table. So all that needs to be done would be to copy the records in the source which do not exist in the destination master table to the destination master table. The same thing could be done for customer and Vendor_Master tables in case the Organisation_Master does not exist in those databases. Proprietary customer numbers and vendor numbers would make this difficult since there could be issues such as two organsations having the same number or a mismatch in their numbers which would have to be resolved.
- In a situation where for business reasons, the company has to have two databases which can be called as X and Y running on different systems and both are operated and updated independant of each other. If the Vendor_Master on Y is updated by adding a few records, these records would be alloted a Vendor Number as per its sequence or algorithm. Later on if these same records are added to the database X the Vendor Number assigned to them is not likely to be the same and so there would be a mismatch between the two databases. Later on if the business feels the need to synchronise the Vendor_Master in both databases such that all vendors in database X are present in database Y, then it would be difficult because there would be mismatches between the records and there could be some which are available in one database but not in another and some may have a Vendor Number assigned which would clash with the Vendor Number assigned to another organisation. An organisation may be assigned one Vendor Number in Database X whereas it may have an entirely different Vendor Number in database Y. Resolving these issues would be a problem. If Standard organisation numbers were available and organisations used them instead of proprietary numbers, then there would be no problem at all in synchronizing the Vendor_Master tables in database X and database Y. All that would need be done would be find out what is missing in each database and then copy these records over to the database table which needs it.
- E-commerce has evolved over the years from what it was in the early years of the internet

to what it is now. It will continue to evolve and the business paradigms of the future could likely be different from what exists today. The business paradigms of the future would most likely find not only standard organisation numbers useful  but all standardised data highly useful as well.

Some of the benefits of standardisation have been mentioned above. However there could perhaps be many more benefits that could be derived from developing standards which may be applicable depending on the kind of organisation and the nature of the bussiness it is involved in.

The MySQL scripts which were used to create the tables in the test data database are shown below. Scripts with the following names are listed in the pages below...

- oldMasters.sql
- oldMasters_rm.sql
- oldMasters_ison.sql
- ison.sql
- drpison.sql

```
-- *-------------------------------------------------------------------------------------*
-- *
-- *    File Name       : oldMasters.sql
-- *                    :
-- *    Function        : Creates the Customer Master and Vendor Master tables in
-- *                    : the traditional format and populates the tables with
-- *                    : sample data.
-- *                    :
-- *    Language        : MySQL DB
-- *    Author          : Vinod Krishnan Menon
-- *
-- *-------------------------------------------------------------------------------------*

CREATE TABLE Customer_Master
(
   Customer_No        Numeric(7) NOT NULL,
   Customer_Name      Char(70)   NOT NULL,
   Address_Ln1        Char(52),
   Address_Ln2        Char(52),
   City_Name          Char(30),
   District_Name      Char(30),
   State_Name         Char(30),
   Country            Char(30),
   PinCode            Char(30)
);

CREATE TABLE Vendor_Master
(
   Vendor_No          Numeric(7) NOT NULL,
   Vendor_Name        Char(70)   NOT NULL,
   Address_Ln1        Char(52),
   Address_Ln2        Char(52),
   City_Name          Char(30),
   District_Name      Char(30),
   State_Name         Char(30),
   Country            Char(30),
   PinCode            Char(30)
);

-- *
-- *----> Add sample records for the tables. Only fields required for demo
-- *----> purposes are populated with data.
-- *

INSERT INTO Customer_Master
   VALUES (6021001, "SmartSystems Inc","","","","","","","" );
```

```
INSERT INTO Customer_Master
   VALUES (6021002, "Om Enterprises","","","","","","","" );
INSERT INTO Customer_Master
   VALUES (6021003, "Precision Electronics Limited","","","","","","","" );

INSERT INTO Vendor_Master
   VALUES (7021044, "SmartSystems Inc","","","","","","","" );
INSERT INTO Vendor_Master
   VALUES (7021045, "SAP AG","","","","","","","" );
INSERT INTO Vendor_Master
   VALUES (7021046, "Microsoft","","","","","","","" );
INSERT INTO Vendor_Master
   VALUES (7021047, "Acer","","","","","","","" );


-- *
-- *----> End Of Script File :  oldMasters.sql
-- *
```

```
-- *--------------------------------------------------------------------------------------*
-- *
-- *    File Name       : oldMasters_rm.sql
-- *                    :
-- *    Function        : Drops the traditional format Customer Master and
-- *                    : Vendor Master tables  from the database so that
-- *                    : it can be replaced with ISON Tables.
-- *                    :
-- *    Language        : MySQL DB
-- *    Author          : Vinod Krishnan Menon
-- *
-- *--------------------------------------------------------------------------------------*

DROP TABLE Customer_Master;
DROP TABLE Vendor_Master;

-- *
-- *----> End Of Script File :  oldMasters_rm.sql
-- *
```

```
-- *-------------------------------------------------------------------------------------*
-- *
-- *    File Name       : oldMasters_ison.sql
-- *                    :
-- *    Function        : Creates the Customer Master and Vendor Master tables
-- *                    : in the traditional format and populates the tables
-- *                    : with sample data.
-- *                    :
-- *    Language        : MySQL DB
-- *    Author          : Vinod Krishnan Menon
-- *
-- *-------------------------------------------------------------------------------------*

CREATE TABLE Customer_Master
(
   ISON               Numeric(12) NOT NULL,
   Customer_Name      Char(70)    NOT NULL,
   Address_Ln1        Char(52),
   Address_Ln2        Char(52),
   City_Name          Char(30),
   District_Name      Char(30),
   State_Name         Char(30),
   Country            Char(30),
   PinCode            Char(30)
);

CREATE TABLE Vendor_Master
(
   ISON               Numeric(12) NOT NULL,
   Vendor_Name        Char(70)    NOT NULL,
   Address_Ln1        Char(52),
   Address_Ln2        Char(52),
   City_Name          Char(30),
   District_Name      Char(30),
   State_Name         Char(30),
   Country            Char(30),
   PinCode            Char(30)
);

-- *
-- *----> Add sample records for the tables. Only fields required for demo
-- *----> purposes are populated with data.
-- *

INSERT INTO Customer_Master
   VALUES (123456000001, "SmartSystems Inc","","","","","","" );
```

```
INSERT INTO Customer_Master
   VALUES (123456000002, "Om Enterprises","","","","","","","" );
INSERT INTO Customer_Master
   VALUES (123456000003, "Precision Electronics Limited","","","","","","","" );

INSERT INTO Vendor_Master
   VALUES (123456000001, "SmartSystems Inc","","","","","","","" );
INSERT INTO Vendor_Master
   VALUES (123456000004, "SAP AG","","","","","","","" );
INSERT INTO Vendor_Master
   VALUES (123456000005, "Microsoft","","","","","","","" );
INSERT INTO Vendor_Master
   VALUES (123456000006, "Acer","","","","","","","" );


-- *
-- *----> End Of Script File :  oldMasters_ison.sql
-- *
```

```
-- *-------------------------------------------------------------------------------------*
-- *
-- *    File Name       : ison.sql
-- *                    :
-- *    Function        : Creates the Organisation_Master and its subordinate
-- *                    : RelationShips table and populates the tables with
-- *                    : sample data to demonstrate the concept.
-- *                    :
-- *    Language        : MySQL DB
-- *    Author          : Vinod Krishnan Menon
-- *
-- *-------------------------------------------------------------------------------------*

CREATE TABLE Organisation_Master
(
   ISON               Numeric(12) NOT NULL,
   Org_Name           Char(70)    NOT NULL,
   Address_Ln1        Char(52),
   Address_Ln2        Char(52),
   City_Name          Char(30),
   District_Name      Char(30),
   State_Name         Char(30),
   Country            Char(30),
   PinCode            Char(6)
);


CREATE TABLE Organisation_Relationships
(
   ISON               Numeric(12) NOT NULL,
   Customer_YN        Char(01),
   Vendor_YN          Char(01),
   SubContractor_YN   Char(01),
   Distributor_YN     Char(01)
);

CREATE VIEW Customer_Master AS
  SELECT
     Organisation_Master.ISON              Customer_No,
     Organisation_Master.Org_Name          Customer_Name,
     Organisation_Master.Address_Ln1,
     Organisation_Master.Address_Ln2,
     Organisation_Master.City_Name,
     Organisation_Master.District_Name,
     Organisation_Master.State_Name,
     Organisation_Master.Country,
```

```
    Organisation_Master.PinCode
   FROM Organisation_Master, Organisation_Relationships
   Where Organisation_Master.ISON = Organisation_Relationships.ISON AND
         Organisation_Relationships.Customer_YN = "Y";


CREATE VIEW Vendor_Master AS
  SELECT
    Organisation_Master.ISON            Vendor_No,
    Organisation_Master.Org_Name        Vendor_Name,
    Organisation_Master.Address_Ln1,
    Organisation_Master.Address_Ln2,
    Organisation_Master.City_Name,
    Organisation_Master.District_Name,
    Organisation_Master.State_Name,
    Organisation_Master.Country,
    Organisation_Master.PinCode
   FROM Organisation_Master, Organisation_Relationships
   Where Organisation_Master.ISON = Organisation_Relationships.ISON AND
         Organisation_Relationships.Vendor_YN = "Y";

CREATE VIEW SubContractor_Master AS
  SELECT
    Organisation_Master.ISON            SubContractor_No,
    Organisation_Master.Org_Name        SubContractor_Name,
    Organisation_Master.Address_Ln1,
    Organisation_Master.Address_Ln2,
    Organisation_Master.City_Name,
    Organisation_Master.District_Name,
    Organisation_Master.State_Name,
    Organisation_Master.Country,
    Organisation_Master.PinCode
   FROM Organisation_Master, Organisation_Relationships
   Where Organisation_Master.ISON = Organisation_Relationships.ISON AND
         Organisation_Relationships.SubContractor_YN = "Y";

CREATE VIEW Distributor_Master AS
  SELECT
    Organisation_Master.ISON            Distributor_No,
    Organisation_Master.Org_Name        Distributor_Name,
    Organisation_Master.Address_Ln1,
    Organisation_Master.Address_Ln2,
    Organisation_Master.City_Name,
    Organisation_Master.District_Name,
    Organisation_Master.State_Name,
    Organisation_Master.Country,
    Organisation_Master.PinCode
```

```
   FROM Organisation_Master, Organisation_Relationships
   Where Organisation_Master.ISON = Organisation_Relationships.ISON AND
         Organisation_Relationships.Distributor_YN = "Y";

-- *
-- *----> Add sample records for the tables. Only fields required for demo
-- *----> purposes are populated with data.
-- *

INSERT INTO Organisation_Master
   VALUES (123456000001, "SmartSystems Inc","","","","","","","" );
INSERT INTO Organisation_Master
   VALUES (123456000002, "Om Enterprises","","","","","","","" );
INSERT INTO Organisation_Master
   VALUES (123456000003, "Precision Electronics Limited","","","","","","","" );
INSERT INTO Organisation_Master
   VALUES (123456000004, "SAP AG","","","","","","","" );
INSERT INTO Organisation_Master
   VALUES (123456000005, "Microsoft","","","","","","","" );
INSERT INTO Organisation_Master
   VALUES (123456000006, "Acer","","","","","","","" );

INSERT INTO Organisation_Relationships VALUES (123456000001,"Y","Y"," "," ");
INSERT INTO Organisation_Relationships VALUES (123456000002,"Y"," "," "," ");
INSERT INTO Organisation_Relationships VALUES (123456000003,"Y"," "," "," ");
INSERT INTO Organisation_Relationships VALUES (123456000004," ","Y"," "," ");
INSERT INTO Organisation_Relationships VALUES (123456000005," ","Y"," "," ");
INSERT INTO Organisation_Relationships VALUES (123456000006," ","Y"," "," ");


-- *
-- *----> End Of Script File :  ison.sql
-- *
```

```
-- *-------------------------------------------------------------------------------------*
-- *
-- *    File Name       : drpison.sql
-- *                    :
-- *    Function        : Drops the ISON Tables and Views from the testdata Database.
-- *                    :
-- *    Language        : MySQL DB
-- *    Author          : Vinod Krishnan Menon
-- *
-- *-------------------------------------------------------------------------------------*

DROP TABLE Organisation_Master;
DROP TABLE Organisation_Relationships;

DROP VIEW Customer_Master;
DROP VIEW Vendor_Master;
DROP VIEW SubContractor_Master;
DROP VIEW Distributor_Master;

-- *
-- *----> End Of Script File :  drpison.sql
-- *
```