ONE QUERY MULTIPLE DATABASES

An interface to mutually access data from heterogeneous databases

Group 5

Abstract:

Migrating large volumes of data from one database storage platform to another is a common problem which is required to be managed to provide the acceptable performance. We elucidate \textit{OQMD} - a common query interface to access data from heterogeneous set of databases. The tool comprises of a \textit{Schema Unification Engine} which takes care of mapping all attributes from one database to another; and a \textit{Common Query Engine} which parses an ``SQL-like'' query and provides access to similar data in different databases. This tool reduces the effort required to search multiple databases with different data mechanics, using a generic approach. The proposed model can be very a helpful and powerful alternative to the time consuming process of ETL(Extract, Transform and Load).

Introduction

In today's era of digitalism, where a number of technologies to maintain and analyze the huge amounts of data have evolved, querying and analyzing data from multiple data repositories with different data mechanics is challenging; esp. when one is RDBM system(like Oracle) and the other is NoSQL system (like HBase). For instance, if a company (which uses HBase) expands its product into a new location, by acquiring a similar product by another company (which uses Oracle) then processing the entire data using a single interface is an intricate task.

The solution which is generally used for the above problem is Data Migration which involves ETL. It is the process of transferring the data between systems or different storage formats, which is a non-trivial task. Complexity of data migration is high which leads to cost overrun and delays with go-live. Whenever a company shifts from one database storage platform to another, they have to undergo the intricate task of migrating data which takes considerable amount of time and effort. Also the learning process of various query languages by the user is a time and cost intensive task.

The proposed solution in this paper simplifies the above two tasks by providing automated mapping technique for all the similar attributes in the databases of the same product, having similar information but not necessarily similar structure using domain mapping and by providing common "SQL-like" API to access data from both datasets simultaneously. With the help of this process, the user can rid of the tedious data migration process altogether and start processing on both the datasets in minutes.

Nishtha[insert content below]

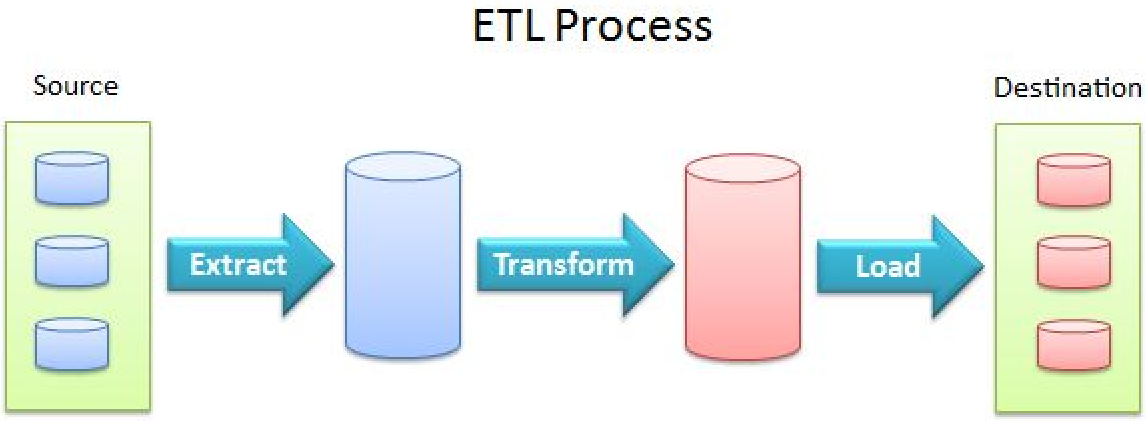
1. ETL
2. HBase
3. RDBMs (Try to include diagrams in all of these; if you copy from somewhere quote that)

ETL(Extract, transform, load)

ETL is short for extract, transform, load, three database functions that are combined into one tool to pull data out of one database and place it into another database.

- Extract is the process of reading data from a database.
- Transform is the process of converting the extracted data from its previous form into the form it needs to be in so that it can be placed into another database. Transformation occurs by using rules or lookup tables or by combining the data with other data.
- Load is the process of writing the data into the target database.

ETL is used to migrate data from one database to another, to form data marts and data warehouses and also to convert databases from one format or type to another.



Performance and Challenges:

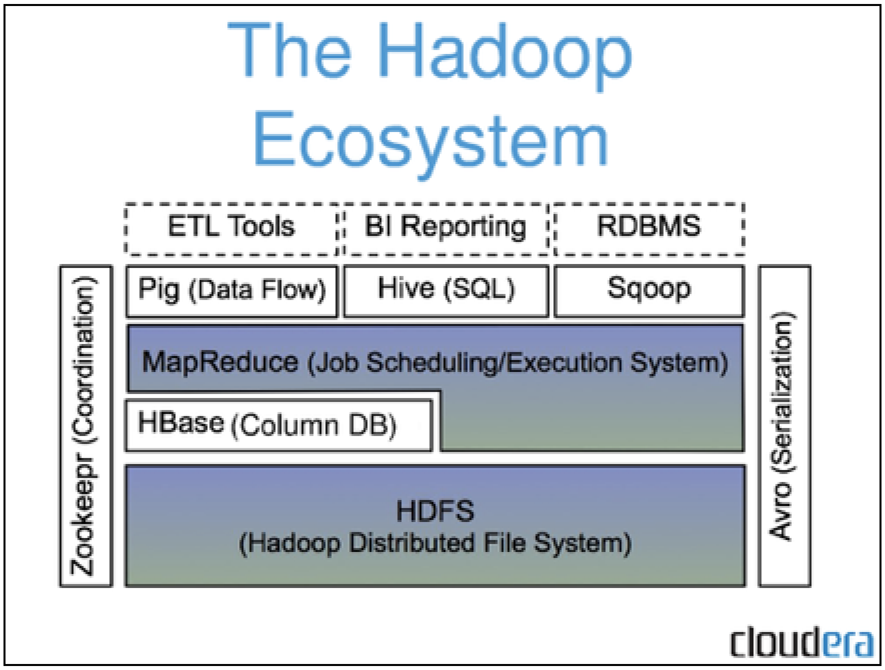
ETL vendors benchmark their record-systems at multiple TB (terabytes) per hour (or ~1 GB per second) using powerful servers with multiple CPUs, multiple hard drives, multiple gigabit-network connections, and lots of memory. The fastest ETL record is currently held by Syncsort,Vertica, and HP at 5.4TB in under an hour, which is more than twice as fast as the earlier record held by Microsoft and Unisys.

ETL processes can involve considerable complexity, and significant operational problems can occur with improperly designed ETL systems. Complexity of ETL processes depends upon the quality of data, Complexity of the Source Data, Meta data, how similar are the source and target data structures etc.

HBASE

HBase is a distributed column-oriented data store built on top of HDFS.It is an Apache open source project whose goal is to provide storage for the Hadoop Distributed Computing.Data is logically organized into tables, rows and columns

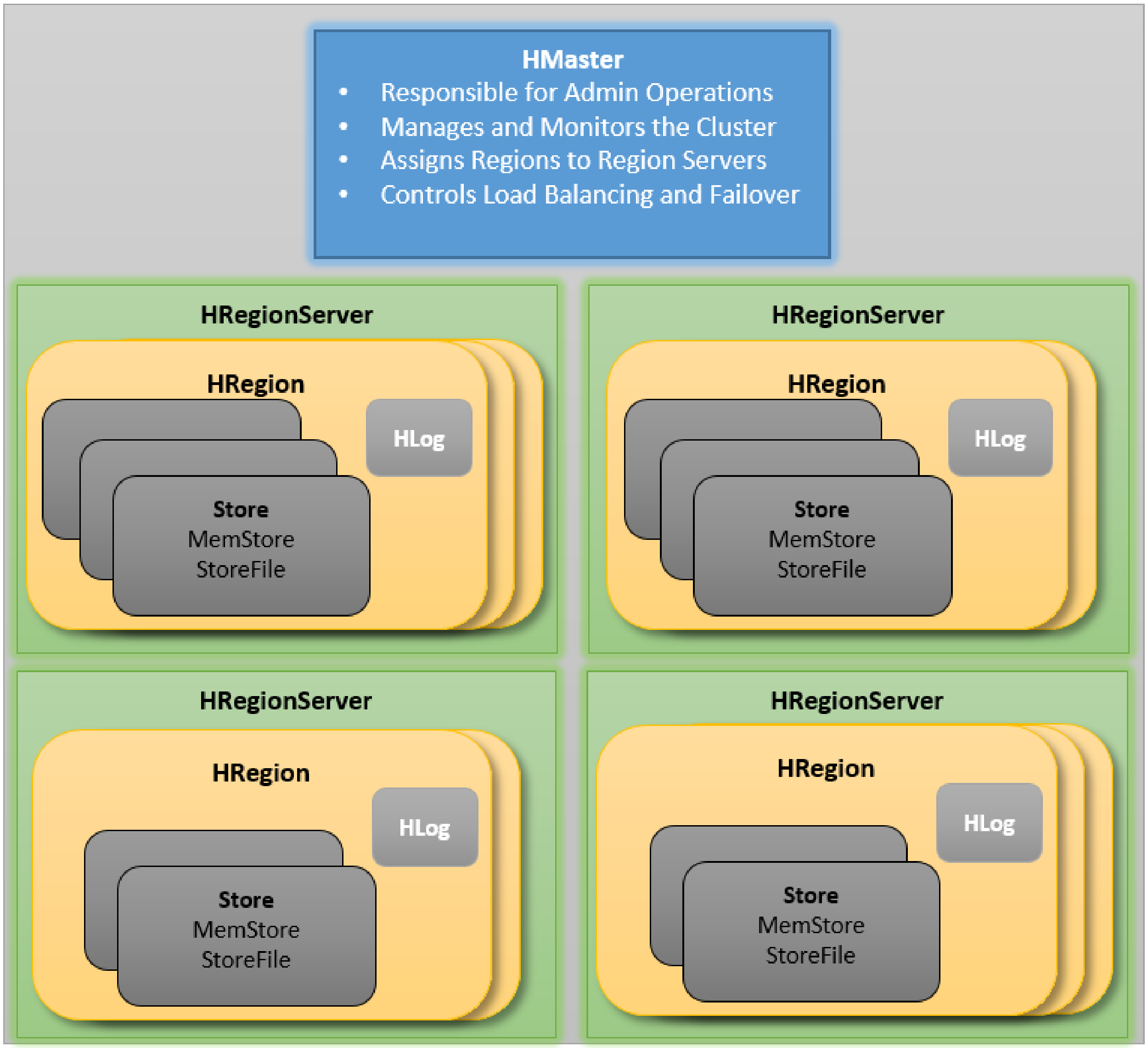
HBase is a column-oriented database that's an open-source implementation of Google's Big Table storage architecture. It can manage structured and semi-structured data and has some built-in features such as scalability, versioning, compression and garbage collection. Since its uses write-ahead logging and distributed configuration, it can provide fault-tolerance and quick recovery from individual server failures. HBase built on top of Hadoop / HDFS and the data stored in HBase can be manipulated using Hadoop's MapReduce capabilities.



Architecture:

The HBase Physical Architecture consists of servers in a Master-Slave relationship as shown below. Typically, the HBase cluster has one Master node, called HMaster and multiple Region Servers called HRegionServer. Each Region Server contains multiple Regions – HRegions.

Just like in a Relational Database, data in HBase is stored in Tables and these Tables are stored in Regions. When a Table becomes too big, the Table is partitioned into multiple Regions. These Regions are assigned to Region Servers across the cluster. Each Region Server hosts roughly the same number of Regions.



The HMaster in the HBase is responsible for

- Performing Administration
- Managing and Monitoring the Cluster

- Assigning Regions to the Region Servers
- Controlling the Load Balancing and Failover

On the other hand, the HRegionServer perform the following work

- Hosting and managing Regions
- Splitting the Regions automatically
- Handling the read/write requests
- Communicating with the Clients directly

Each Region Server contains a Write-Ahead Log (called HLog) and multiple Regions. Each Region in turn is made up of a MemStore and multiple StoreFiles (HFile). The data lives in these StoreFiles in the form of Column Families (explained below). The MemStore holds in-memory modifications to the Store (data).

The mapping of Regions to Region Server is kept in a system table called .META. When trying to read or write data from HBase, the clients read the required Region information from the .META table and directly communicate with the appropriate Region Server. Each Region is identified by the start key (inclusive) and the end key (exclusive)

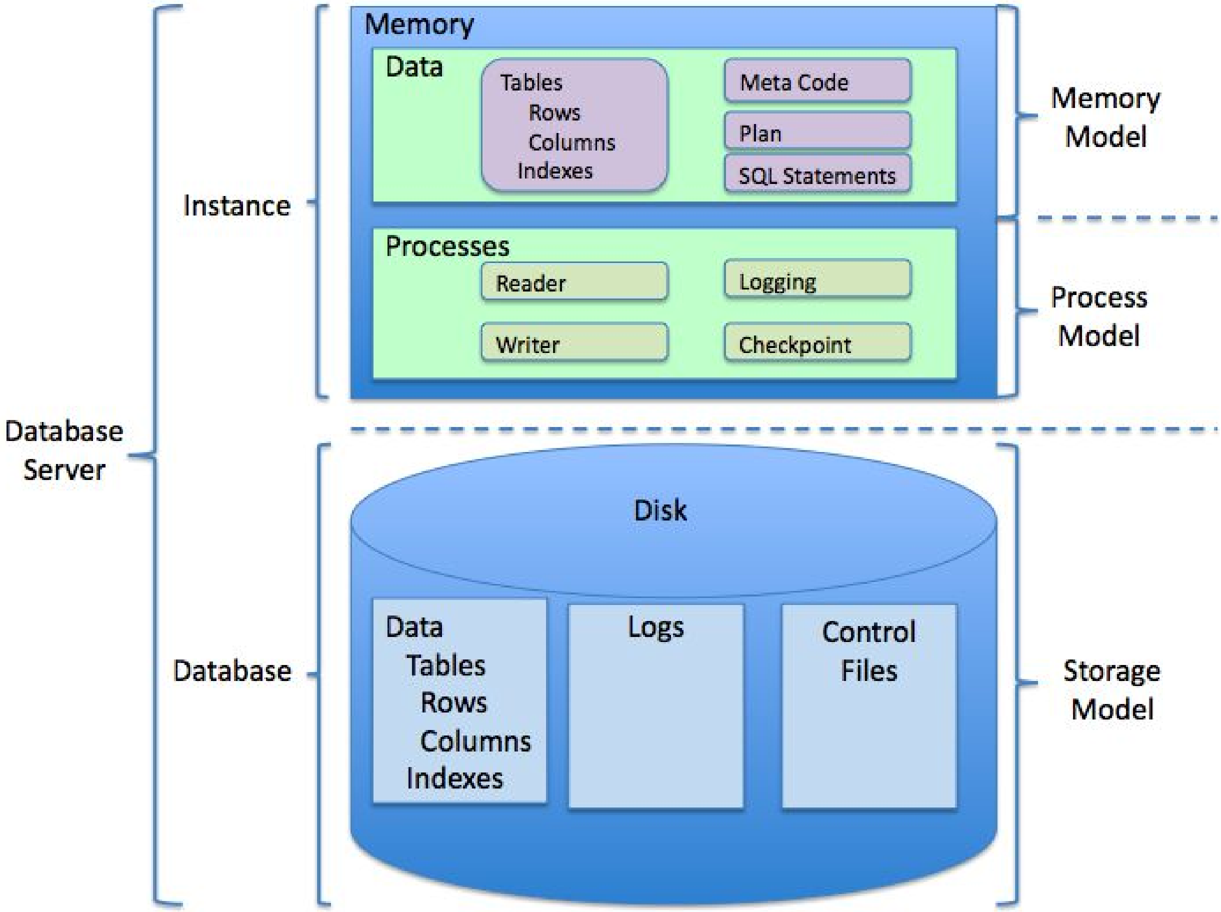
Features:

- Linear and modular scalability.
- Strictly consistent reads and writes.
- Automatic and configurable sharding of tables
- Automatic failover support between RegionServers.
- Convenient base classes for backing Hadoop MapReduce jobs with Apache HBase tables.
- Easy to use Java API for client access.
- Block cache and Bloom Filters for real-time queries.
- Query predicate push down via server side Filters
- Thrift gateway and a REST-ful Web service that supports XML, Protobuf, and binary data encoding options
- Extensible jruby-based (JIRB) shell
- Support for exporting metrics via the Hadoop metrics subsystem to files or Ganglia; or via JMX

<u>RDBMS</u>

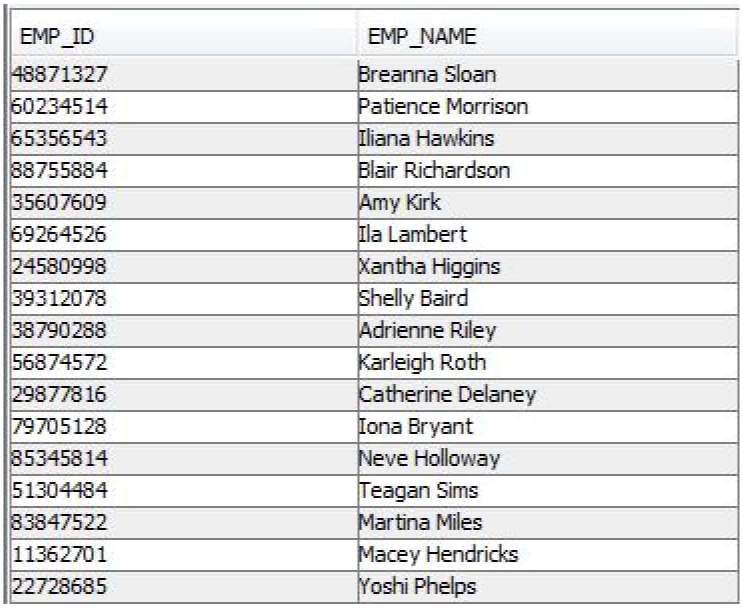
A relational database management system (RDBMS) is a database management system (DBMS) that is based on therelational model as invented by E. F. Codd, of IBM's San Jose Research Laboratory. Many popular databases currently in use are based on the relational database model.

RDBMSs are a common choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and other applications since the 1980s. Relational databases have often replaced legacyhierarchical databases and network databases because they are easier to understand and use. However, relational databases have received unsuccessful challenge attempts by object database management systems in the 1980s and 1990s (which were introduced trying to address the so-called object-relational impedance mismatch between relational databases and object-oriented application programs) and also by XML database management systems in the 1990s.[citation needed] Despite such attempts, RDBMSs keep most of the market share, which has also grown over the years.



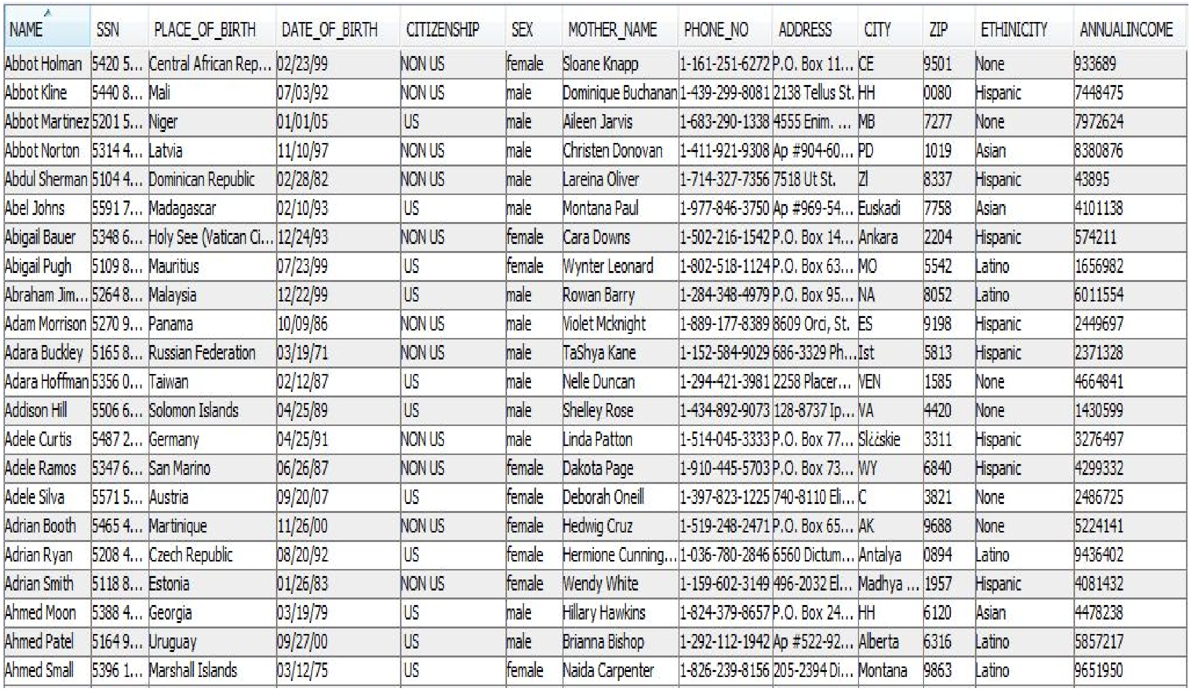
OQMD accuracy:

For simple cases like where databases have columns belonging to unique patterns, For example the table shown below have emp_id and emp_name which both belongs to different pattern classes of number and string respectively. For these type of cases we are able to achieve the 100% accuracy.



EMP_ID	EMP_NAME
48871327	Breanna Sloan
60234514	Patience Morrison
65356543	Iliana Hawkins
88755884	Blair Richardson
35607609	Amy Kirk
69264526	Ila Lambert
24580998	Xantha Higgins
39312078	Shelly Baird
38790288	Adrienne Riley
56874572	Karleigh Roth
29877816	Catherine Delaney
79705128	Iona Bryant
85345814	Neve Holloway
51304484	Teagan Sims
83847522	Martina Miles
11362701	Macey Hendricks
22728685	Yoshi Phelps

But for close to real life scenarios 85% accuracy is achieved by our tool. For example for the table we used for our testing purpose (generated from “http://generatedata.com/”)given below Name and Mother_name attributes belongs to the same pattern class which will fails the result of automating mapping and reduce the efficiency. For this type of scenario we need to enter mapping information manually.



NAME	SSN	PLACE_OF_BIRTH	DATE_OF_BIRTH	CITIZENSHIP	SEX	MOTHER_NAME	PHONE_NO	ADDRESS	CITY	ZIP	ETHINICITY	ANNUALINCOME
Abbot Holman	5420 5...	Central African Rep...	02/23/99	NON US	female	Sloane Knapp	1-161-251-6272	P.O. Box 11...	CE	9501	None	933689
Abbot Kline	5440 8...	Mali	07/03/92	NON US	male	Dominique Buchanan	1-439-299-8081	2138 Tellus St.	HH	0080	Hispanic	7448475
Abbot Martinez	5201 5...	Niger	01/01/05	US	male	Aileen Jarvis	1-683-290-1338	4555 Enim. ...	MB	7277	None	7972624
Abbot Norton	5314 4...	Latvia	11/10/97	NON US	male	Christen Donovan	1-411-921-9308	Ap #904-60...	PD	1019	Asian	8380876
Abdul Sherman	5104 4...	Dominican Republic	02/28/82	NON US	male	Lareina Oliver	1-714-327-7356	7518 Ut St.	ZI	8337	Hispanic	43895
Abel Johns	5591 7...	Madagascar	02/10/93	US	male	Montana Paul	1-977-846-3750	Ap #969-54...	Euskadi	7758	Asian	4101138
Abigail Bauer	5348 6...	Holy See (Vatican Ci...	12/24/93	NON US	female	Cara Downs	1-502-216-1542	P.O. Box 14...	Ankara	2204	Hispanic	574211
Abigail Pugh	5109 8...	Mauritius	07/23/99	US	female	Wynter Leonard	1-802-518-1124	P.O. Box 63...	MO	5542	Latino	1656982
Abraham Jim...	5264 8...	Malaysia	12/22/99	US	male	Rowan Barry	1-284-348-4979	P.O. Box 95...	NA	8052	Latino	6011554
Adam Morrison	5270 9...	Panama	10/09/86	NON US	male	Violet Mcknight	1-889-177-8389	8609 Orci, St.	ES	9198	Hispanic	2449697
Adara Buckley	5165 8...	Russian Federation	03/19/71	NON US	male	TaShya Kane	1-152-584-9029	686-3329 Ph...	Ist	5813	Hispanic	2371328
Adara Hoffman	5356 0...	Taiwan	02/12/87	US	male	Nelle Duncan	1-294-421-3981	2258 Placer...	VEN	1585	None	4664841
Addison Hill	5506 6...	Solomon Islands	04/25/89	US	male	Shelley Rose	1-434-892-9073	128-8737 Ip...	VA	4420	None	1430599
Adele Curtis	5487 2...	Germany	04/25/91	NON US	male	Linda Patton	1-514-045-3333	P.O. Box 77...	Sl¿¿skie	3311	Hispanic	3276497
Adele Ramos	5347 6...	San Marino	06/26/87	NON US	female	Dakota Page	1-910-445-5703	P.O. Box 73...	WY	6840	Hispanic	4299332
Adele Silva	5571 5...	Austria	09/20/07	US	female	Deborah Oneill	1-397-823-1225	740-8110 Eli...	C	3821	None	2486725
Adrian Booth	5465 4...	Martinique	11/26/00	NON US	female	Hedwig Cruz	1-519-248-2471	P.O. Box 65...	AK	9688	None	5224141
Adrian Ryan	5208 4...	Czech Republic	08/20/92	US	female	Hermione Cunning...	1-036-780-2846	6560 Dictum...	Antalya	0894	Latino	9436402
Adrian Smith	5118 8...	Estonia	01/26/83	NON US	female	Wendy White	1-159-602-3149	496-2032 El...	Madhya ...	1957	Hispanic	4081432
Ahmed Moon	5388 4...	Georgia	03/19/79	US	male	Hillary Hawkins	1-824-379-8657	P.O. Box 24...	HH	6120	Asian	4478238
Ahmed Patel	5164 9...	Uruguay	09/27/00	US	male	Brianna Bishop	1-292-112-1942	Ap #522-92...	Alberta	6316	Latino	5857217
Ahmed Small	5396 1...	Marshall Islands	03/12/75	US	female	Naida Carpenter	1-826-239-8156	205-2394 Di...	Montana	9863	Latino	9651950

Conclusion:

The purpose of this project is to demonstrate a methodology for building an SQL-like query interface which can uniformly run over multiple databases of heterogeneous nature. Two main problems have been worked upon. Firstly, the problem of de-multiplexing the query into different databases has been solved by creating the *Common Query Engine* which parses, decodes and fetches results from different databases.

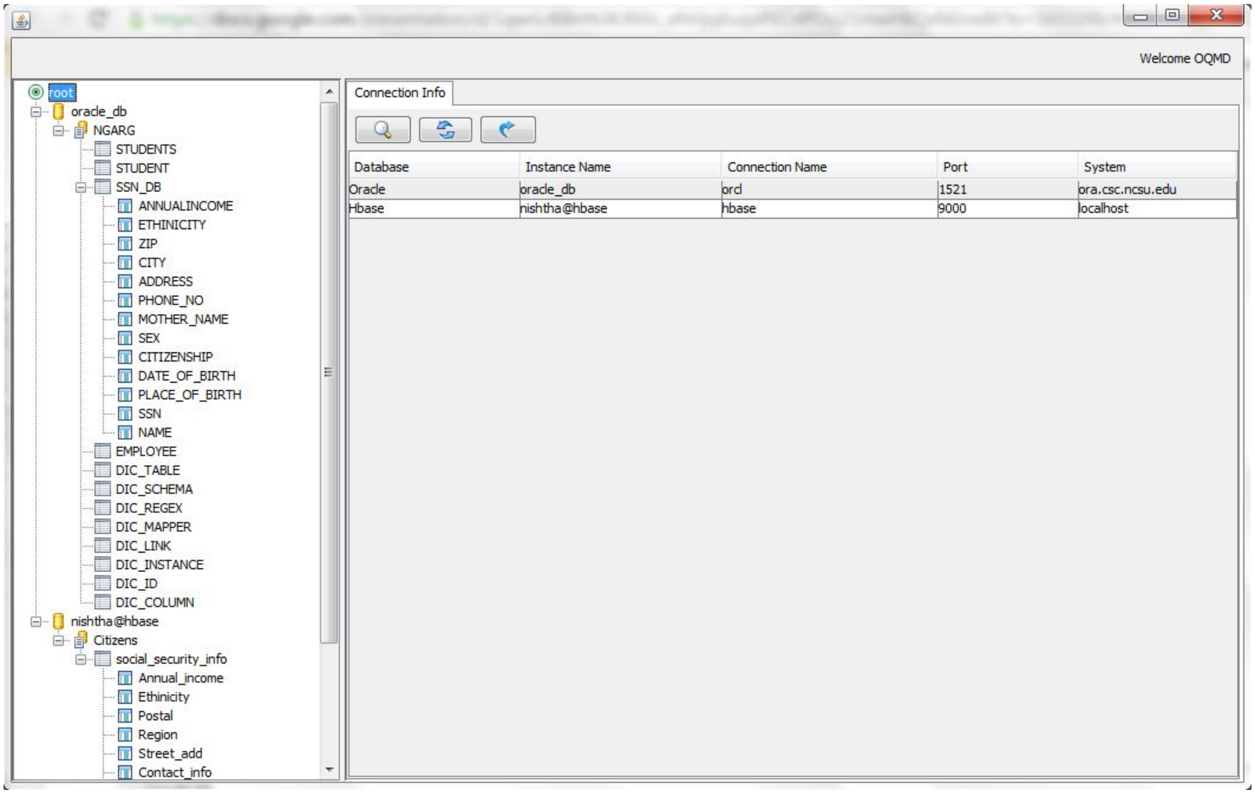
Secondly, the problem of identifying potential mappings for different attributes/columns in different databases has been addressed by creating a *Schema Unification Engine*. This engine identifies the patterns of the values of a particular attribute and classifies and maps similar attributes based on this classification. This problem has a vast scope and the performance of attribute mapping largely depends on the completeness of the Regex and Link databases. We have created a prototype of this approach to demonstrate the correctness and feasibility of this methodology.

Arnab[insert content below]

1. DB Crawler (with screenshots of all the features)
2. Meta database
3. Gudusoft parser
4. Create a demo with screen shots
5. Overview of the Algorithm

DB Crawler:

DB Crawler is a data discovery tool which can be used to create connections over any database and see its contents such as schemas, tables, columns and their metadata. When a user creates a database connection it is persistent as all the metadata of the corresponding connection is stored in the Meta Database using a unique ID.



Welcome OQMD

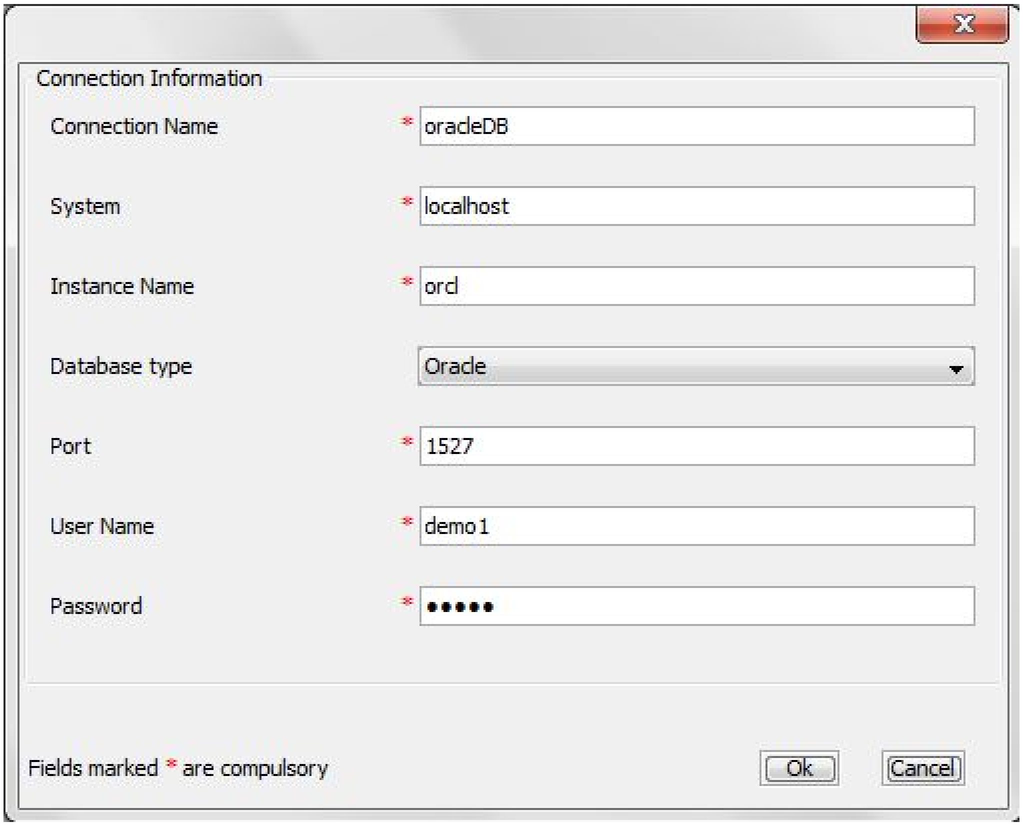
- root
  - oracle_db
    - NGARG
      - STUDENTS
      - STUDENT
      - SSN_DB
        - ANNUALINCOME
        - ETHINICITY
        - ZIP
        - CITY
        - ADDRESS
        - PHONE_NO
        - MOTHER_NAME
        - SEX
        - CITIZENSHIP
        - DATE_OF_BIRTH
        - PLACE_OF_BIRTH
        - SSN
        - NAME
      - EMPLOYEE
      - DIC_TABLE
      - DIC_SCHEMA
      - DIC_REGEX
      - DIC_MAPPER
      - DIC_LINK
      - DIC_INSTANCE
      - DIC_ID
      - DIC_COLUMN
  - nishtha@hbase
    - Citizens
      - social_security_info
        - Annual_income
        - Ethinicity
        - Postal
        - Region
        - Street_add
        - Contact_info

Connection Info

Database	Instance Name	Connection Name	Port	System
Oracle	oracle_db	orcl	1521	ora.csc.ncsu.edu
Hbase	nishtha@hbase	hbase	9000	localhost

DB Crawler showing the Left Hierarchy and the Right View **[img 1]**

The DB Crawler view is separated horizontally as shown in **img 1**. The left part is a tree of connections, schemas, tables and columns which is referred to as Left Hierarchy in the paper. The right part shows the detailed view of the node of the tree on which the user clicks which is referred to as Right View in the paper. For example if a user clicks on a Table Node on the Left Hierarchy, the Right View will display the table's metadata and the data in different tabs. There is also an optional Mapper tab which is shown if the columns of the table have been mapped to other columns in a second table in another database.



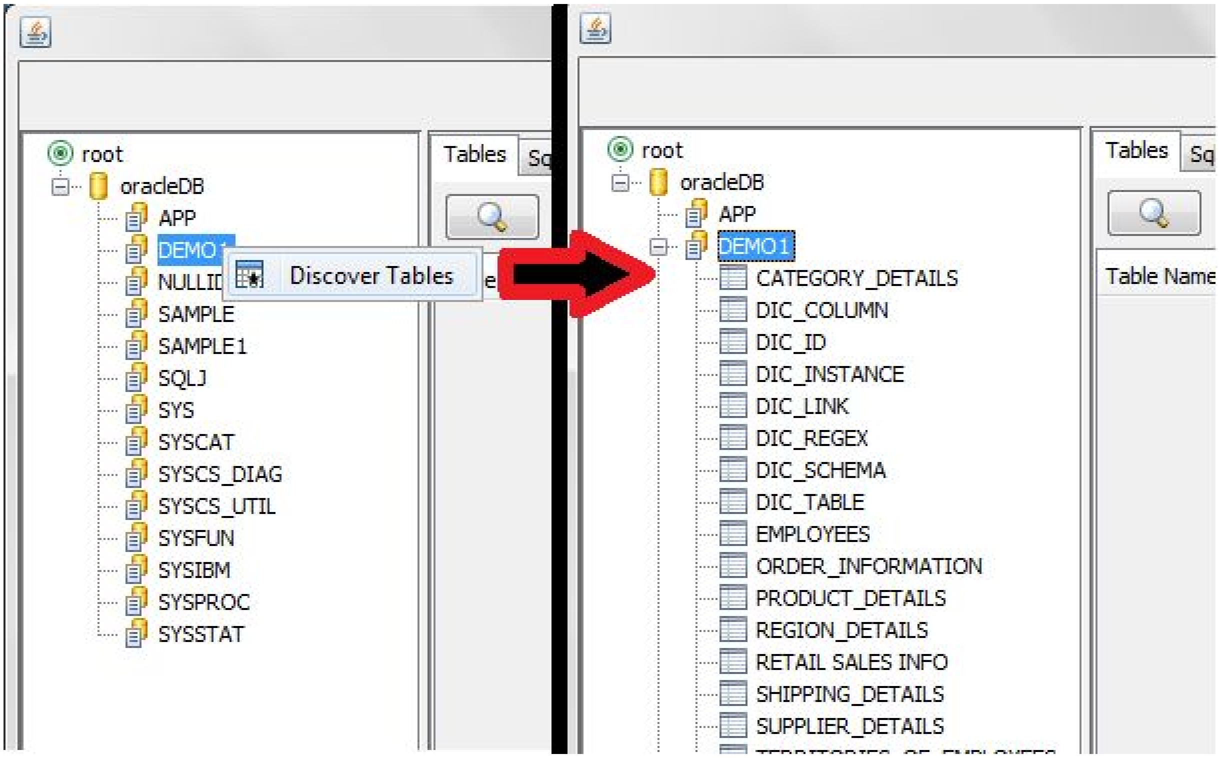
Connection Information

Connection Name *	oracleDB
System *	localhost
Instance Name *	orcl
Database type	Oracle
Port *	1527
User Name *	demo1
Password *	●●●●●

Fields marked * are compulsory

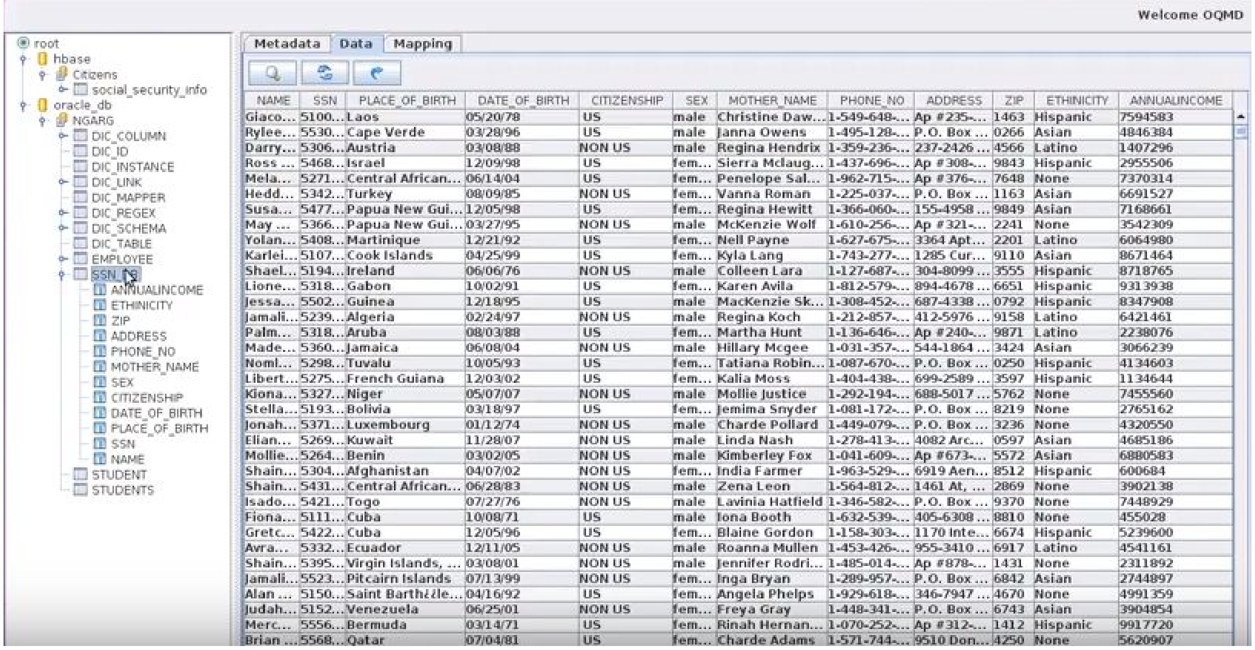
Add Connection **[img 2]**

A first time user will start by creating a connection to a database by providing the correct parameters **[img 2]**. On successful creation of the connection a node with the connection name will appear on the Left Hierarchy. Then the user can discover schemas on the connection which will show all the schemas contained in the connection. The user can further discover tables and discover columns similarly **[img 3]**.



Discover tables on Demo1 **[img 3]**

The user can view the metadata and data of a table on the Right View in different tabs as shown in [**img 4**]. Additionally the user can run Profile Columns on a table which compares the contents of all the columns to standard regular expressions and try to understand the contents of the columns. It also supports Map Table where the user can specify that the columns in this table should be mapped to the columns in another table.



NAME	SSN	PLACE_OF_BIRTH	DATE_OF_BIRTH	CITIZENSHIP	SEX	MOTHER_NAME	PHONE_NO	ADDRESS	ZIP	ETHINICITY	ANNUALINCOME
Giaco...	5100...	Laos	05/20/78	US	male	Christine Daw...	1-549-648-...	Ap #235-...	1463	Hispanic	7594583
Rylee...	5530...	Cape Verde	03/28/96	US	male	Janna Owens	1-495-128-...	P.O. Box ...	0266	Asian	4846384
Darry...	5306...	Austria	03/08/88	NON US	male	Regina Hendrix	1-359-236-...	237-2426 ...	4566	Latino	1407296
Ross ...	5468...	Israel	12/09/98	US	fem...	Sierra Mclaug...	1-437-696-...	Ap #308-...	9843	Hispanic	2955506
Mela...	5271...	Central African...	06/14/04	US	fem...	Penelope Sal...	1-962-715-...	Ap #376-...	7648	None	7370314
Hedd...	5342...	Turkey	08/09/85	NON US	fem...	Vanna Roman	1-225-037-...	P.O. Box ...	1163	Asian	6691527
Susa...	5477...	Papua New Gui...	12/05/98	US	fem...	Regina Hewitt	1-366-060-...	155-4958 ...	9849	Asian	7168661
May ...	5366...	Papua New Gui...	03/27/95	NON US	male	McKenzie Wolf	1-610-256-...	Ap #321-...	2241	None	3542309
Yolan...	5408...	Martinique	12/21/92	US	fem...	Nell Payne	1-627-675-...	3364 Apt...	2201	Latino	6064980
Karlei...	5107...	Cook Islands	04/25/99	US	fem...	Kyla Lang	1-743-277-...	1285 Cur...	9110	Asian	8671464
Shael...	5194...	Ireland	06/06/76	NON US	male	Colleen Lara	1-127-687-...	304-8099 ...	3555	Hispanic	8718765
Lione...	5318...	Gabon	10/02/91	US	fem...	Karen Avila	1-812-579-...	894-4678 ...	6651	Hispanic	9313938
Jessa...	5502...	Guinea	12/18/95	US	male	MacKenzie Sk...	1-308-452-...	687-4338 ...	0792	Hispanic	8347908
Jamali...	5239...	Algeria	02/24/97	NON US	male	Regina Koch	1-212-857-...	412-5976 ...	9158	Latino	6421461
Palm...	5318...	Aruba	08/03/88	US	fem...	Martha Hunt	1-136-646-...	Ap #240-...	9871	Latino	2238076
Made...	5360...	Jamaica	06/08/04	NON US	male	Hillary Mcgee	1-031-357-...	544-1864 ...	3424	Asian	3066239
Noml...	5298...	Tuvalu	10/05/93	US	fem...	Tatiana Robin...	1-087-670-...	P.O. Box ...	0250	Hispanic	4134603
Libert...	5275...	French Guiana	12/03/02	US	fem...	Kalia Moss	1-404-438-...	699-2589 ...	3597	Hispanic	1134644
Kiona...	5327...	Niger	05/07/07	NON US	male	Mollie Justice	1-292-194-...	688-5017 ...	5762	None	7455560
Stella...	5193...	Bolivia	03/18/97	US	fem...	Jemima Snyder	1-081-172-...	P.O. Box ...	8219	None	2765162
Jonah...	5371...	Luxembourg	01/12/74	NON US	male	Charde Pollard	1-449-079-...	P.O. Box ...	3236	None	4320550
Elian...	5269...	Kuwait	11/28/07	NON US	male	Linda Nash	1-278-413-...	4082 Arc...	0597	Asian	4685186
Mollie...	5264...	Benin	03/02/05	NON US	male	Kimberley Fox	1-041-609-...	Ap #673-...	5572	Asian	6880583
Shain...	5304...	Afghanistan	04/07/02	NON US	fem...	India Farmer	1-963-529-...	6919 Aen...	8512	Hispanic	600684
Shain...	5431...	Central African...	06/28/83	NON US	male	Zena Leon	1-564-812-...	1461 At, ...	2869	None	3902138
Isado...	5421...	Togo	07/27/76	NON US	male	Lavinia Hatfield	1-346-582-...	P.O. Box ...	9370	None	7448929
Fiona...	5111...	Cuba	10/08/71	US	male	Iona Booth	1-632-539-...	405-6308 ...	8810	None	455028
Gretc...	5422...	Cuba	12/05/96	US	fem...	Blaine Gordon	1-158-303-...	1170 Inte...	6674	Hispanic	5239600
Avra...	5332...	Ecuador	12/11/05	NON US	male	Roanna Mullen	1-453-426-...	955-3410 ...	6917	Latino	4541161
Shain...	5395...	Virgin Islands, ...	03/08/01	NON US	male	Jennifer Rodri...	1-485-014-...	Ap #878-...	1431	None	2311892
Jamali...	5523...	Pitcairn Islands	07/13/99	NON US	fem...	Inga Bryan	1-289-957-...	P.O. Box ...	6842	Asian	2744897
Alan ...	5150...	Saint Barth¿¿le...	04/16/92	US	fem...	Angela Phelps	1-929-618-...	346-7947 ...	4670	None	4991359
Judah...	5152...	Venezuela	06/25/01	NON US	fem...	Freya Gray	1-448-341-...	P.O. Box ...	6743	Asian	3904854
Merc...	5556...	Bermuda	03/14/71	US	fem...	Rinah Hernan...	1-070-252-...	Ap #312-...	1412	Hispanic	9917720
Brian ...	5568...	Qatar	07/04/81	US	fem...	Charde Adams	1-571-744-...	9510 Don...	4250	None	5620907

Meta Database:

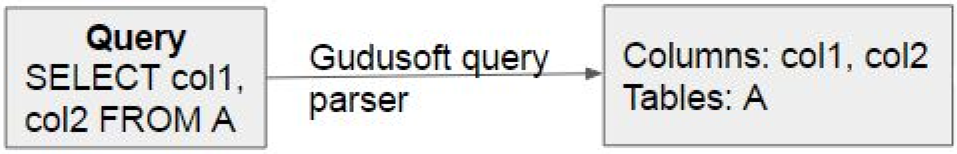
In our project we have used an Oracle database for our metadata management which is referred to as Meta Database. The Meta Database consists of eight tables as named below:

1. DIC_INSTANCE
2. DIC_SCHEMA
3. DIC_TABLE
4. DIC_COLUMN
5. DIC_ID
6. DIC_REGEX
7. DIC_LINK
8. DIC_MAPPER

The first four tables contain metadata of the connections, schemas, tables and columns respectively. The fifth table is used to generate new IDs for the project. The sixth table contains the common regular expressions; for example postal code, phone numbers, etc. This table is used for profiling of columns in a table so that each column can be mapped to a particular regular expression. The seventh table is a helper table for DIC_REGEX. Some data do not correspond to a regular expression but they belong to a range of values. For example type gender can contain two values: male and female; type countries can belong to a range of values. These kinds of values are stored in the DIC_LINK table and they share a one-to-many relationship with DIC_REGEX table. The eighth table is used to store the results from the Map Table. The columns in the first table which could be mapped to another column in the second are stored for querying from the common query engine. Additionally it stores the efficiency of the match which gives the user an empirical estimate of the mapping.

Gudusoft SQL parser:

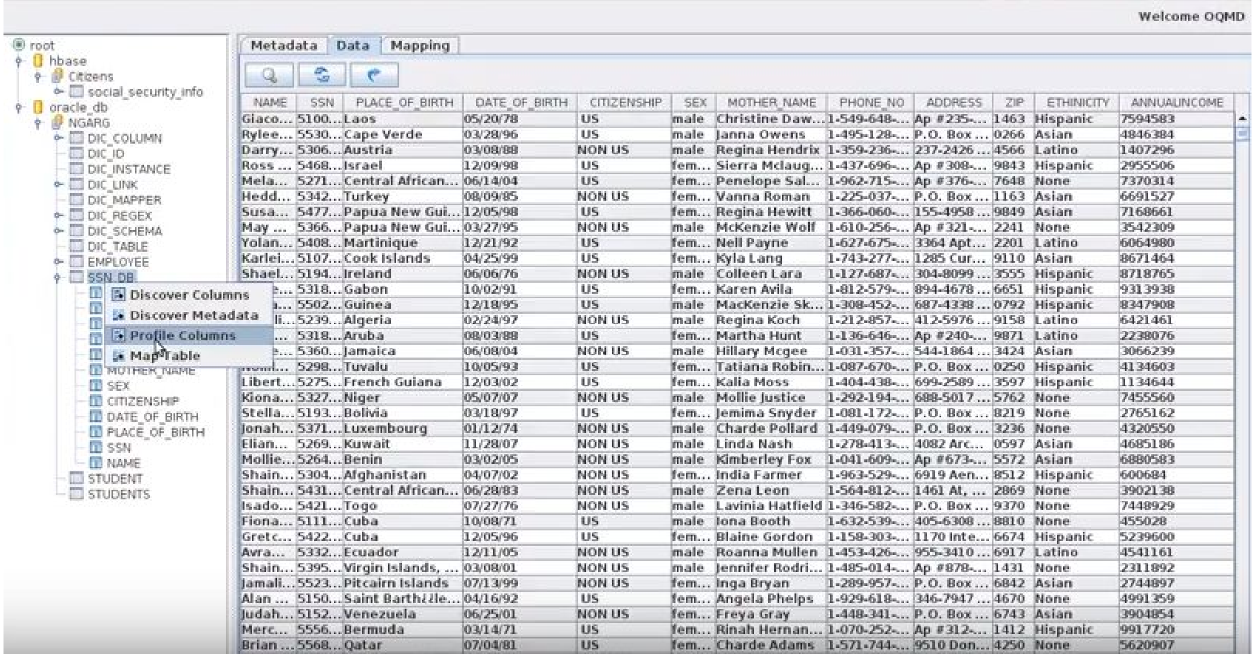
This is a Java library which can parse any SQL to fetch the different parts such column names, table names, where clause, sort by, etc. Due to the complexity of the SQL grammar we have used the SQL parser to help us parse complex SQL as well. The common query engine on receiving a query first runs it through the parser to fetch the column names, table names, schema names and where clause. This is the first step in the Common Query Engine.



Working of Gudusoft SQL parser **[img 4]**

Demo with screenshots:

The Schema Unification Engine profiles all the columns of a table and assigns a domain to each column. This feature is done by right clicking on the table and selecting Profile Columns as shown in **[img 5]**. The profiling should be done to a similar table in another database as well so that the mapping algorithm can run on both the tables.



NAME	SSN	PLACE_OF_BIRTH	DATE_OF_BIRTH	CITIZENSHIP	SEX	MOTHER_NAME	PHONE_NO	ADDRESS	ZIP	ETHINICITY	ANNUALINCOME
Giaco...	5100...	Laos	05/20/78	US	male	Christine Daw...	1-549-648-...	Ap #235-...	1463	Hispanic	7594583
Rylee...	5530...	Cape Verde	03/28/96	US	male	Janna Owens	1-495-128-...	P.O. Box ...	0266	Asian	4846384
Darry...	5306...	Austria	03/08/88	NON US	male	Regina Hendrix	1-359-236-...	237-2426 ...	4566	Latino	1407296
Ross ...	5468...	Israel	12/09/98	US	fem...	Sierra Mclaug...	1-437-696-...	Ap #308-...	9843	Hispanic	2955506
Mela...	5271...	Central African...	06/14/04	US	fem...	Penelope Sal...	1-962-715-...	Ap #376-...	7648	None	7370314
Hedd...	5342...	Turkey	08/09/85	NON US	fem...	Vanna Roman	1-225-037-...	P.O. Box ...	1163	Asian	6691527
Susa...	5477...	Papua New Gui...	12/05/98	US	fem...	Regina Hewitt	1-366-060-...	155-4958 ...	9849	Asian	7168661
May ...	5366...	Papua New Gui...	03/27/95	NON US	male	McKenzie Wolf	1-610-256-...	Ap #321-...	2241	None	3542309
Yolan...	5408...	Martinique	12/21/92	US	fem...	Nell Payne	1-627-675-...	3364 Apt...	2201	Latino	6064980
Karlei...	5107...	Cook Islands	04/25/99	US	fem...	Kyla Lang	1-743-277-...	1285 Cur...	9110	Asian	8671464
Shael...	5194...	Ireland	06/06/76	NON US	male	Colleen Lara	1-127-687-...	304-8099 ...	3555	Hispanic	8718765
e...	5318...	Gabon	10/02/91	US	fem...	Karen Avila	1-812-579-...	894-4678 ...	6651	Hispanic	9313938
a...	5502...	Guinea	12/18/95	US	male	MacKenzie Sk...	1-308-452-...	687-4338 ...	0792	Hispanic	8347908
li...	5239...	Algeria	02/24/97	NON US	male	Regina Koch	1-212-857-...	412-5976 ...	9158	Latino	6421461
...	5318...	Aruba	08/03/88	US	fem...	Martha Hunt	1-136-646-...	Ap #240-...	9871	Latino	2238076
e...	5360...	Jamaica	06/08/04	NON US	male	Hillary Mcgee	1-031-357-...	544-1864 ...	3424	Asian	3066239
[illegible]	5298...	Tuvalu	10/05/93	US	fem...	Tatiana Robin...	1-087-670-...	P.O. Box ...	0250	Hispanic	4134603
Libert...	5275...	French Guiana	12/03/02	US	fem...	Kalia Moss	1-404-438-...	699-2589 ...	3597	Hispanic	1134644
Kiona...	5327...	Niger	05/07/07	NON US	male	Mollie Justice	1-292-194-...	688-5017 ...	5762	None	7455560
Stella...	5193...	Bolivia	03/18/97	US	fem...	Jemima Snyder	1-081-172-...	P.O. Box ...	8219	None	2765162
Jonah...	5371...	Luxembourg	01/12/74	NON US	male	Charde Pollard	1-449-079-...	P.O. Box ...	3236	None	4320550
Elian...	5269...	Kuwait	11/28/07	NON US	male	Linda Nash	1-278-413-...	4082 Arc...	0597	Asian	4685186
Mollie...	5264...	Benin	03/02/05	NON US	male	Kimberley Fox	1-041-609-...	Ap #673-...	5572	Asian	6880583
Shain...	5304...	Afghanistan	04/07/02	NON US	fem...	India Farmer	1-963-529-...	6919 Aen...	8512	Hispanic	600684
Shain...	5431...	Central African...	06/28/83	NON US	male	Zena Leon	1-564-812-...	1461 At, ...	2869	None	3902138
Isado...	5421...	Togo	07/27/76	NON US	male	Lavinia Hatfield	1-346-582-...	P.O. Box ...	9370	None	7448929
Fiona...	5111...	Cuba	10/08/71	US	male	Iona Booth	1-632-539-...	405-6308 ...	8810	None	455028
Gretc...	5422...	Cuba	12/05/96	US	fem...	Blaine Gordon	1-158-303-...	1170 Inte...	6674	Hispanic	5239600
Avra...	5332...	Ecuador	12/11/05	NON US	male	Roanna Mullen	1-453-426-...	955-3410 ...	6917	Latino	4541161
Shain...	5395...	Virgin Islands, ...	03/08/01	NON US	male	Jennifer Rodri...	1-485-014-...	Ap #878-...	1431	None	2311892
Jamali...	5523...	Pitcairn Islands	07/13/99	NON US	fem...	Inga Bryan	1-289-957-...	P.O. Box ...	6842	Asian	2744897
Alan ...	5150...	Saint Barth¿le...	04/16/92	US	fem...	Angela Phelps	1-929-618-...	346-7947 ...	4670	None	4991359
Judah...	5152...	Venezuela	06/25/01	NON US	fem...	Freya Gray	1-448-341-...	P.O. Box ...	6743	Asian	3904854
Merc...	5556...	Bermuda	03/14/71	US	fem...	Rinah Hernan...	1-070-252-...	Ap #312-...	1412	Hispanic	9917720
Brian ...	5568...	Qatar	07/04/81	US	fem...	Charde Adams	1-571-744-...	9510 Don...	4250	None	5620907

Profile Columns on SSN_DB **[img 5]**

After the profiling has been done to two tables, the mapping algorithm can be run. This is done by right clicking on the table and selecting Map Table. The function of Map Table is to relate a table from one database to another table in another database. Only the Profiled Tables can be mapped to one another. The Schema Unification Engine assumes that all the columns are already assigned to a domain and has a Regex ID associated with it. The algorithm will try to match the columns in both the tables which belong to the same domain, i.e. columns having same Regex IDs. The user can view all the columns that belong to the same domain from the Mapping View as shown in **img 6**.

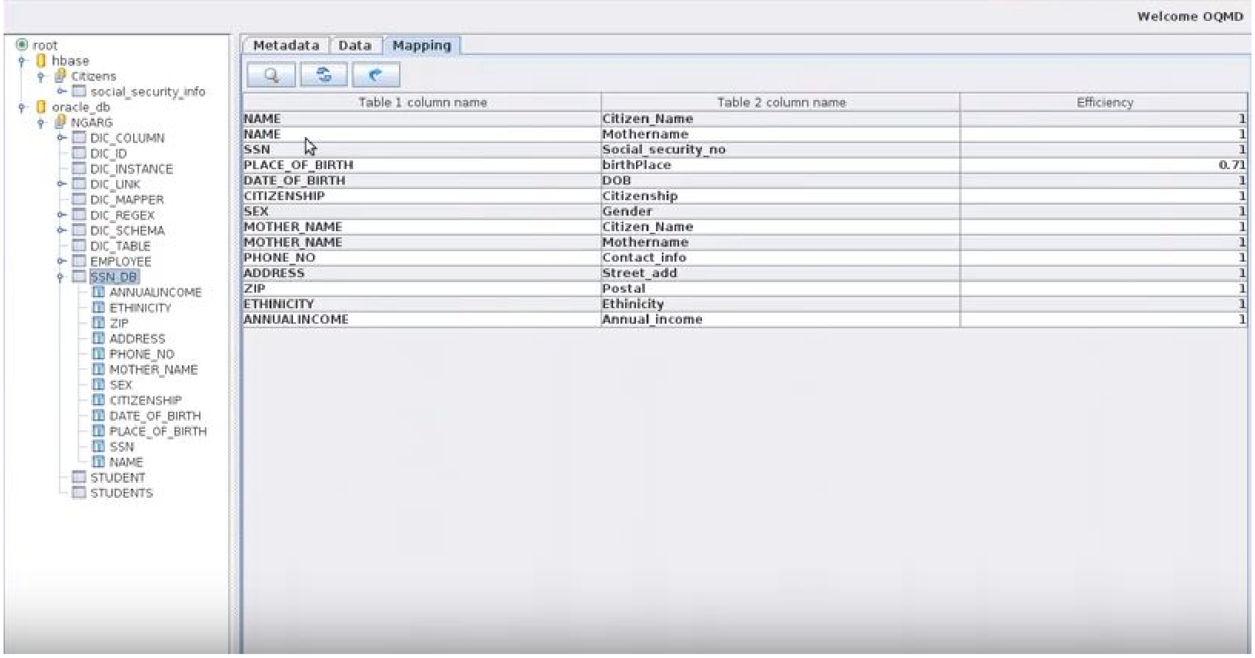
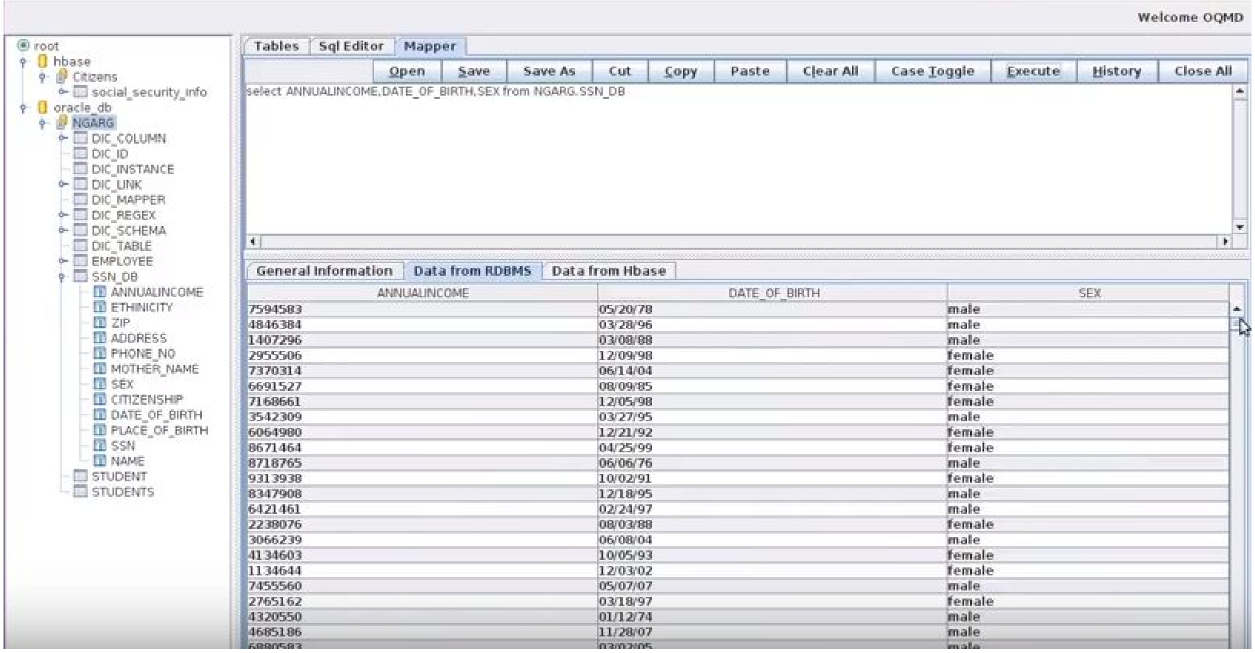


Table 1 column name	Table 2 column name	Efficiency
NAME	Citizen_Name	1
NAME	Mothername	1
SSN	Social_security_no	1
PLACE_OF_BIRTH	birthPlace	0.71
DATE_OF_BIRTH	DOB	1
CITIZENSHIP	Citizenship	1
SEX	Gender	1
MOTHER_NAME	Citizen_Name	1
MOTHER_NAME	Mothername	1
PHONE_NO	Contact_info	1
ADDRESS	Street_add	1
ZIP	Postal	1
ETHINICITY	Ethinicity	1
ANNUALINCOME	Annual_income	1

Mapping View **[img 6]**

If two columns match to a single domain, the algorithm cannot figure out which columns are the true match. As in **img 6**, the column NAME matches with the columns Citizen_Name and Mothername because both of the columns match to the same domain. Then the algorithm fails and manual mapping is needed for those columns. The algorithm maps the columns for which there were no conflicts. Then the user can use the Mapper View to query on the first table and the Single Query Interface will generate the subsequent query to fetch the corresponding columns from the other table as shown in **img 7.** The data from both the tables will be shown in multiple tabs and shown in **img 7**.



```
select ANNUALINCOME,DATE_OF_BIRTH,SEX from NGARG.SSN_DB
```

ANNUALINCOME	DATE_OF_BIRTH	SEX
7594583	05/20/78	male
4846384	03/28/96	male
1407296	03/08/88	male
2955506	12/09/98	female
7370314	06/14/04	female
6691527	08/09/85	female
7168661	12/05/98	female
3542309	03/27/95	male
6064980	12/21/92	female
8671464	04/25/99	female
8718765	06/06/76	male
9313938	10/02/91	female
8347908	12/18/95	male
6421461	02/24/97	male
2238076	08/03/88	female
3066239	06/08/04	male
4134603	10/05/93	female
1134644	12/03/02	female
7455560	05/07/07	male
2765162	03/18/97	female
4320550	01/12/74	male
4685186	11/28/07	male
6880583	03/02/05	male

Mapper SQL View **[img 7]**

In **img 7**, the data from RDBMS is shown in the second tab and data from Hbase is shown in the third tab. The user has entered the query to fetch data from SSN_DB and the Single Query Interface has fetched the data from both the tables SSN_DB and social_security_info.

Overview of the algorithm:

Arjun[insert content below]

Algorithm:
Db Crawler → Attribute Extraction → Column Profiling → Attribute Mapping → Schema Unification

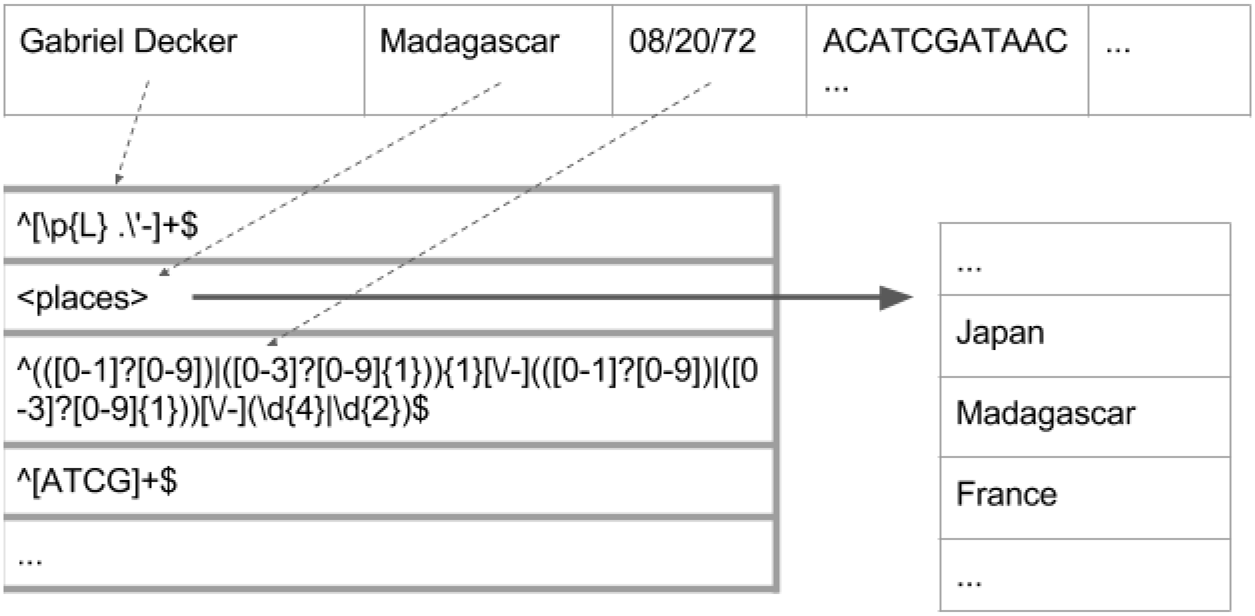
Assumptions:

Mapping is being performed between databases which have data pertaining to the same use-case.

Column Profiling : The main goal of this step is to segregate all the attributes into *domains*. We define a domain to be a classification for any textual data pertaining to the pattern it exhibits. This pattern maybe a regex or link pattern.

- **Regex** (Regular Expression) is a sequence of characters that defines a generic pattern. For instance, an attribute like "Name" will match with a pattern like *^[\p{L} .'-]+$* or an attribute like "DNA sequence" will match with a pattern like *^[ATCG]+$*.
- **Link** refers to the category of data which can be mapped (or linked) to a certain predefined values. We predict these values by using a dictionary reverse indexed by the

broad category in which any word/phrase can be classified into. For instance, an attribute like “Disease_name” can have values like *Cholera*, *Typhoid*, *Meningitis*, etc. and such values can be matched using the dictionary data pertaining to the domain “disease”.



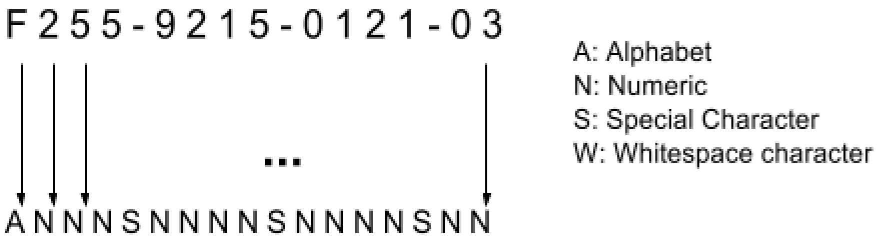
For performing this classification, we extract random samples (for each attribute/column) of 100 values from the entire population(data) from each database, and find the confidence level (CL = the number of matches/100) with which a particular attribute can be classified into a particular domain. If the CL is less than 0.9(chosen after trial and experimentation) then another sample is taken. This process is followed again and again and each time a cumulative CL is computed. This process continues until either we achieve a cumulative CL greater than 0.9 or 10 samples have been exhausted.

Also, it has been attempted to order all the *regex* and *link* data from more specific to more generalized categories and therefore column profiling is done in this order. This is so because a regex pattern for “Name” can also match to the pattern of “DNA sequence” and hence is a more generalized pattern. Thus, when an attribute is matched to a more specific pattern then it need not be matched further for generalized domains. Adopting the same argument, we match for *link* data before matching for *regex* data.

Schema Unification: This step involves mapping of each attribute from one database to another. Using column profiling we can narrow down the attribute from being any arbitrary value to something significant. Now, have some knowledge about the attributes before we move on to mapping them with potentially similar attributes in different databases. At this step we assume that each attribute is classified into some domain. This may lead to two cases:

- There could be only one attribute from each database in each domain. In such a case, a unique mapping can be achieved directly, for those attributes.

- There could be more than one attribute from each database in each domain. In such a case, each tuple *t=<attr a,attr b, ...>*(in which each attribute belongs to a different database) of attributes in each domain is run through the following procedure:
    - Character-type pattern: For each attribute in the tuple the corresponding pattern of character-type is matched. For instance, an attribute “Driver_License_No” may have a value like *F255-9215-0121-03* and the corresponding character pattern as shown in the figure. This pattern is matched for each attribute in the tuple.



    - For numerical attributes, we can deploy certain statistical approach to distinguish and map them. Currently, we take the mean of the difference between 50 values of each pair of attributes, and this is repeated for 20 samples. The cumulative mean is compared and the pair with the least mean is mapped.
      This method works to distinguish between numerical attributes with large difference between them but it may not work in counter-cases. Therefore, in order to map numerical attributes more and stronger statistics like like quantiles, variance, mean, etc. can be used. Statistic approaches like z-test, linear regression, etc. can also be used for more accurate mapping.

    - SOP tagger : not sure…..

With the aforementioned methodology, we get the mappings for each attribute in a database to every other attribute in other databases. However, there maybe cases of reduced accuracy i.e. where some attributes may not get mapped to any attribute or some attributes may get mapped to more than one attributes. Such cases have been discussed in …..

Common query engine: When an “SQL-like” query like

SELECT a1,a2 FROM table1 WHERE <condition>

is entered in the interface, it is process using the following methodology:

- Parsing: In order to extract different components of the SQL-like query, it is parsed using gudusoft.gsqlparser(cite:http://www.sqlparser.com/kb/javadoc/gudusoft/gsqlparser/package-summary.html). This would segregate the contents of the query and return as:

```
getResultColumnList() → a1,a2
getTables() → table1
getWhereClause() → <condition>
...
```

- Forwarding: The original query and the parsed components are forwarded to database specific worker APIs (which sit over the each database separately). These workers fetch the attribute mapping information and further call the corresponding APIs or execute the corresponding queries in the databases to access their data. Once the results from the individual databases are prepared, they are forwarded back via these workers to the user where the results are shown cumulatively.
  As mentioned earlier, currently we have support for two databases namely-Oracle(RDBMS) and HBase. Therefore, we have two worker APIs:
  RDBMS worker API: As the source query is a subset of the SQL thus, RDBMS implicitly supports the query entered by the user. Therefore, the worker API simply executes the same query using the native *execute* calls for SQL, in JAVA. In case, the table name used in the query belongs to Hbase, the worker API will find the corresponding mapping for the attributes in the query, from the meta-database and further re-create the query with native attributes and table name.
  Hbase worker API:
  Elaborate on worker APIs for RDBMS and HBase.

Figures from ppt are to be used in all sections