

Maksym Iaroshenko

Co-Founder and Senior Software Engineer at Eltrino

Magento non-MySQL implementations

<http://ice.eltrino.com/>

MySQL ?

- Magento OOB supports MySQL only
- Since release of Magento CE 1.6 and Magento EE 1.11 it could potentially work with other RDBMS (e.g. Microsoft SQL Server, Oracle, PostgreSQL)
- It was announced by Andrey Tserkus (a member of Magento team) back in 2011 at Magento Developers Paradise

Magento
Developers Paradise

PayPal



Intro

- Magento experience since 2007
- One of former Magento developers
- Was working on RDBMS implementation into Magento
- Working on large MSSQL project
- Starting another Oracle project

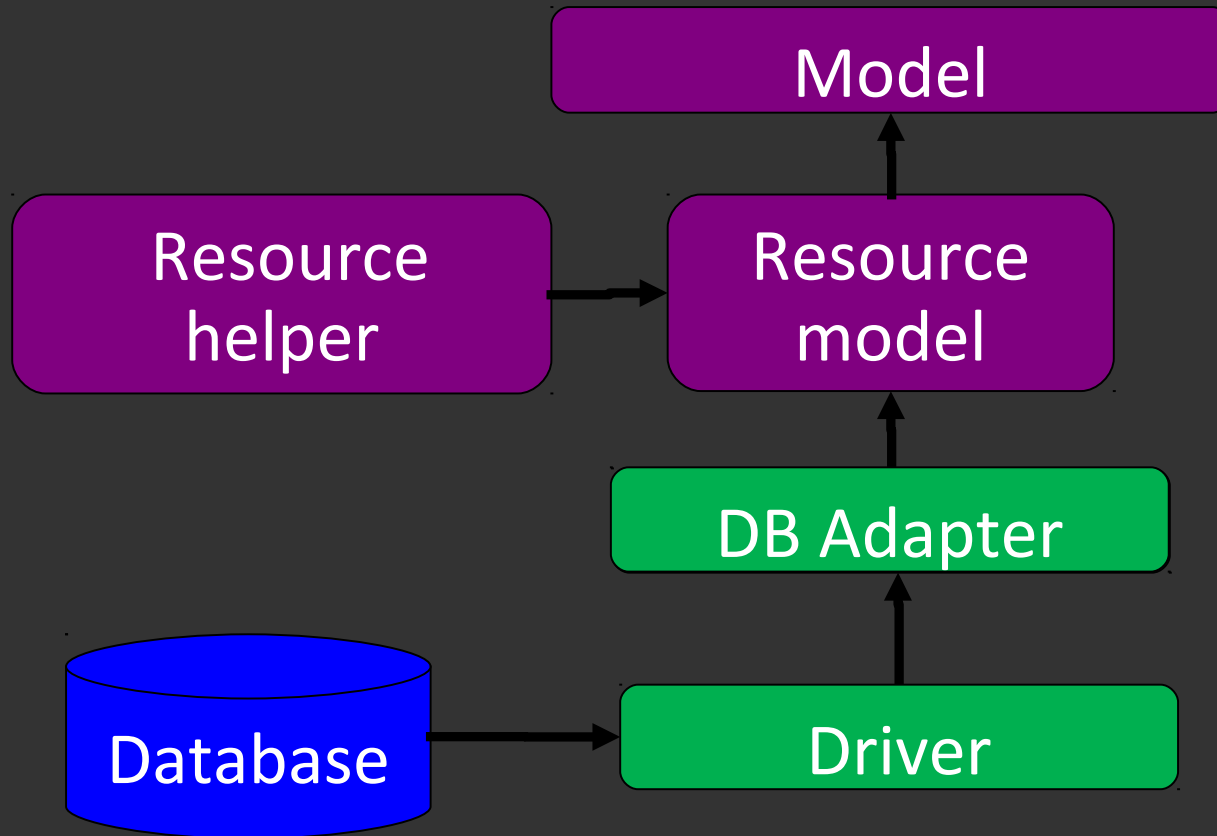
Why not MySQL ?

- Security standards
- SLA
- Support
- Corporate policies
- Technical advantages, potential performance increase because of using other RDBMS features
- etc

Agenda

- How DB layer works in Magento?
- What is the role of resource helpers?
- How to support new Database?
- Common problems & solutions
- Available features in other RDBMS

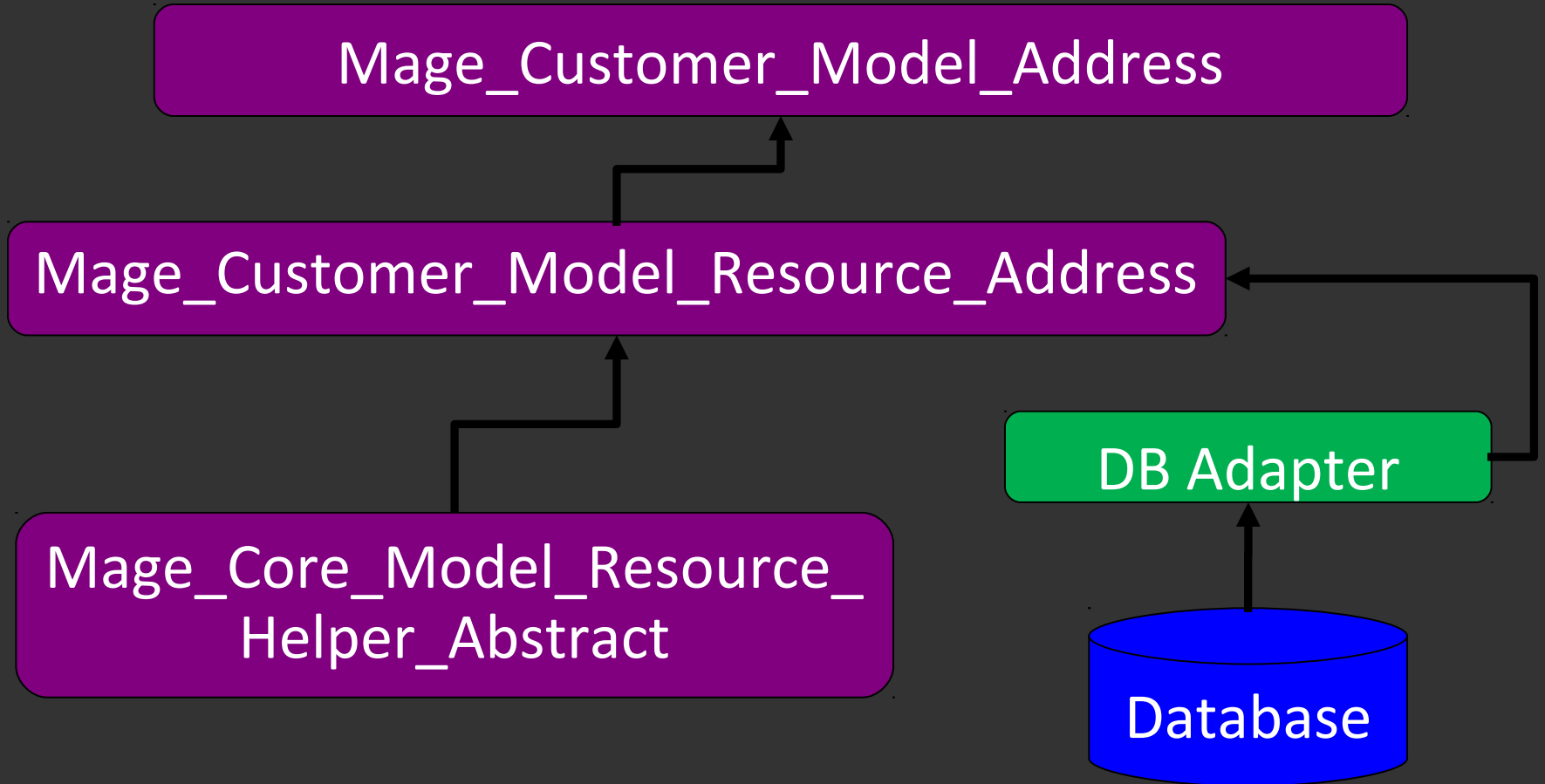
How DB layer works in Magento



Responsibilities of DB Adapter

- Connect to database
- Perform platform related operations
- Provide platform related queries
- Management of nested transactions

Example of interaction



Example of interaction

```
Class Mage_Core_Model_Resource_Helper_Mssql
    extends Mage_Core_Model_Resource_Helper_Abstract
{
    .....
    public function groupConcatPrepare
    (Varien_Db_Select $select, $groupConcatDelimiter = ',')
    {
    .....
    return $select; //Varien_Db_Select
```

Example of interaction

```
Class Mage_Customer_Model_Resource_Address
  extends ...

{

  public function getName()
  {
    .....
    $select = Mage::getResourceHelper('core')
      ->groupConcatPrepare($this->getSelect())
    $this->_getReadAdapter()->fetchRow($select);
    .....
  }
}
```

The role of resource helpers

Not all differences between databases can be covered by adapter. Primary role of resource helper is to get the same result of execution for resource models.

Example:

MySQL

```
SELECT GROUP_CONCAT(field)  
FROM table
```

Oracle

```
SELECT LISTAGG(field)  
FROM table
```

Example of res/helper usage

- Usage in resource model:

```
$select->where(  
    Mage::getResourceHelper('catalog')  
        ->getIsNullNotNullCondition('pis.value', 'pid.value'));
```

- Method of resource helper

```
public function getIsNullNotNullCondition($field1, $field2)  
{  
    return sprintf('%s IS NOT NULL',  
        $this->_getReadAdapter()  
            ->getIfNullSql($field1, $field2));  
}
```

How to support new Database?

- Create new adapter which will implements native Magento Db Adapter Interface
- Define Adapter class in specific resource model & configuration
- Prefer PDO db libs, because they have a lot common parts with existing adapter

Enabling new DB layer

- Create a new adapter class
Varien_Db_Adapter_Xrdbms
- Create a new statement class (if needed)
Varien_Db_Statement_Xrdbms
- Create a new resource type DB
Mage_Core_Model_Resource_Type_Db_Xrdbms
- Create a new resource helper
Mage_Core_Model_Resource_Helper_Xrdbms

Define Adapter class in app/etc/config.xml

```
<config>
  <global>
    <resource>
      <connection>
        <types>
          <pdo_xrdbms>
            <adapter>Varien_Db_Adapter_Pdo_xRdbms</adapter>
            <class>
              Mage_Core_Model_Resource_Type_Db_Pdo_xRdbms
            </class>
          </pdo_xrdbms>
        </types>
      </connection>
    </resource>
```

.....

Connection in app/etc/local.xml

```
<config>
  <global>
    <default_setup>
      <connection>
        <host><![CDATA[192.168.0.1]]></host>
        <username><![CDATA[sa]]></username>
        <password><![CDATA[qwerty]]></password>
        <dbname><![CDATA[dbname]]></dbname>
        <initStatements><![CDATA[]]></initStatements>
        <model><![CDATA[xrdms]]></model>
        <type><![CDATA[pdo_xrdbms]]></type>
        <pdoType><![CDATA[dblib]]></pdoType>
        <active>1</active>
      </connection>
    </default_setup>
  .....
```

Common problems & solutions

- Differences in transactions
- Differences in SQL syntax

Differences in transactions

- Almost all databases have some differences in transactions logic
- Before starting developing a new adapter be sure that you understand this differences

Differences in SQL syntax

- The main problem on the way to support new DB is differences in SQL syntax. All database have a differences in their dialects.
- Next slides show frequently uses cases:

Aggregation

MySQL :

```
SELECT * FROM catalog_product_entity AS entity
INNER JOIN catalog_product_index_price AS price ON
price.entity_id = entity.entity_id
GROUP BY entity.entity_id
```

SQL Server :

```
SELECT entity.entity_id, MAX(entity.sku) AS sku,
MAX(price.final_price) AS max_final_price FROM
catalog_product_entity AS entity
INNER JOIN catalog_product_index_price price ON
price.entity_id = entity.entity_id
GROUP BY entity.entity_id
```

GROUP_CONCAT

MySQL:

```
SELECT GROUP_CONCAT(value SEPARATOR ' ') AS name  
FROM customer_entity_varchar  
WHERE attribute_id in (5,7)  
GROUP BY entity_id;
```

SQL Server:

```
SELECT stuff((SELECT cast(' ' as varchar(max)) +  
v.value  
                FROM customer_entity_varchar v  
                WHERE v.entity_id = cev.entity_id AND  
attribute_id in (5,7)  
                for xml path('')), 1, 1, '') AS name  
FROM customer_entity_varchar AS cev  
GROUP BY entity_id;
```

DATETIME manipulation

MySQL:

```
SELECT (TO_DAYS('2012-12-12 00:00:00') -  
TO_DAYS('2012-12-10 00:00:00'));
```

SQL Server:

```
SELECT DATEDIFF(DAY, '2012-12-10 00:00:00', '2012-12-  
12 00:10:00');
```


LIMIT

MySQL:

```
SELECT * FROM customer_entity LIMIT 5,10;
```

SQL Server:

```
SELECT * FROM (  
    SELECT TOP (5 + 10) * , ROW_NUMBER() OVER ( ORDER  
BY code) AS rownum  
    FROM customer_entity ) AS t  
WHERE rownum > 5
```

Available features in other RDBMS

- Analytic functions (Anywhere but not in MySQL)
- Materialized view (Oracle, DB-2)
- OLAP functions (PIVOT, UNPIVOT)

Analytic functions

- Analytic functions compute an aggregate value based on a group of rows. They differ from aggregate functions in the way that they return multiple rows for each group
- Can be used for Reports, Indexers, anywhere where we need compute statistic

Materialized view

- A materialized view is a database object that contains the results of a query. It may be join result, or summary based on aggregations of a table's data
- Can be used for Reports, Indexers, anywhere where we need get calculated data by one click.

PIVOT

- You can use the PIVOT and UNPIVOT relational operators to change a table valued expression into another table.
- Can be used for EAV based reports.

Reference

[http://www.magentocommerce.com/blog
/comments/creating-adapters-for-rbdms/](http://www.magentocommerce.com/blog/comments/creating-adapters-for-rbdms/)

Q&A

maksym@eltrino.com