

Drupal databases TNG

Taking the database
up a notch.



Larry "Crell" Garfield
Senior Programmer,
Palantir.net

PHP 5 Zend Certified
Engineer



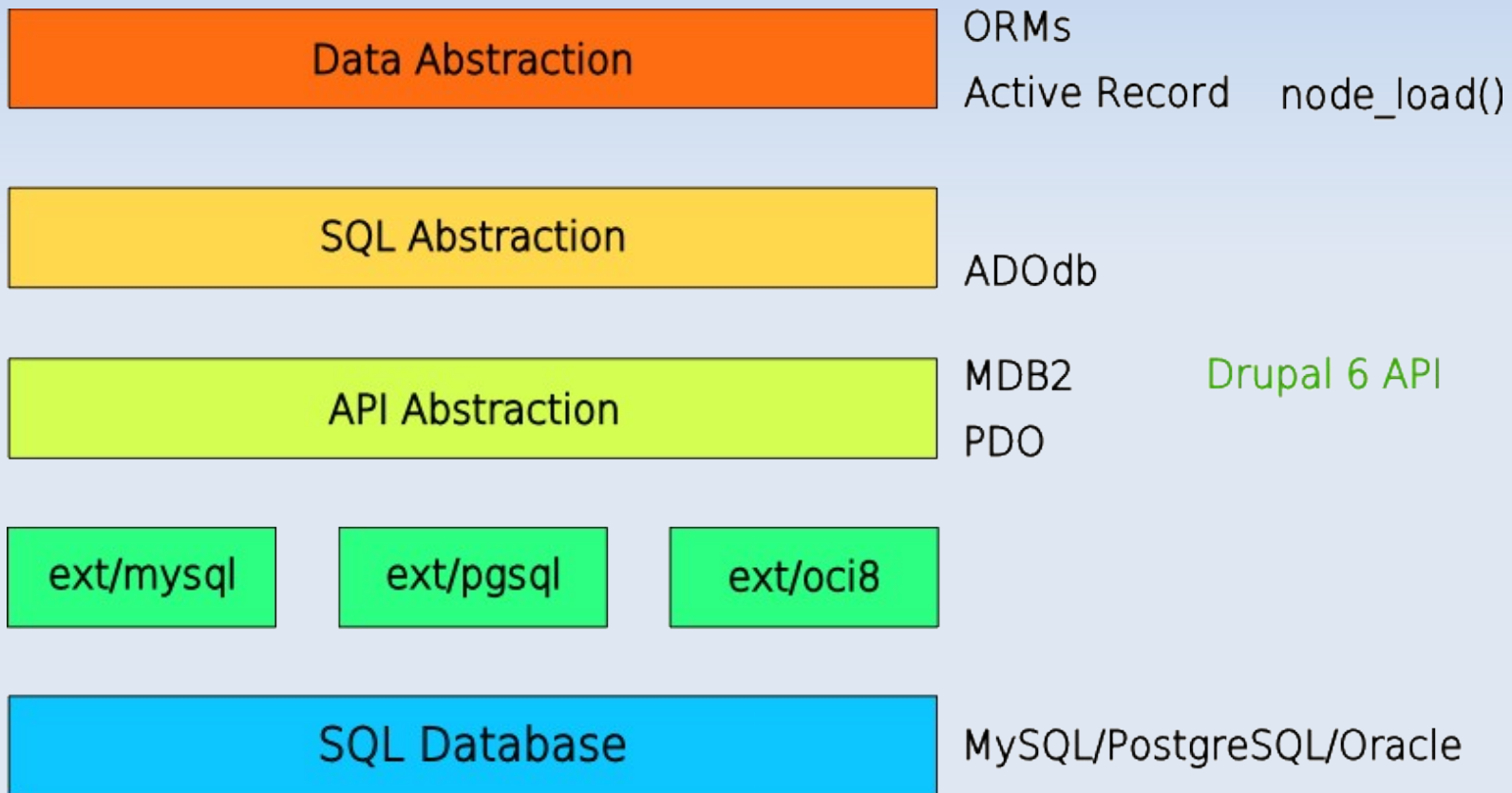
API potpourri

- ext/mysql
 - MySQL 3-based
 - Not in base distribution \geq PHP 5.0
- ext/mysqli
 - MySQL 4.1-based
 - Not in base distribution
- ext/pgsql
 - Not in base distribution

PDO

- Unified API
- In base distribution \geq PHP 5.1
- Features
 - OOP API
 - Type-agnostic prepared statements
- DBL, not DBAL

Database layer cake



API abstraction (PDO)

```
$db = new PDO('mysql:host=localhost;dbname=test', $user, $pass);  
$res = $db->query("SELECT * FROM node WHERE uid=1");  
while ($record = $res->fetch()) { echo $record->nid; }
```

```
$statement = $db->prepare("SELECT * FROM node where uid=?");  
$statement->execute(array(1));  
foreach ($statement as $record) {  
    echo $record->nid;  
}  
$statement->execute(array(2));
```

...

SQL abstraction

- Simple abstraction: Query level
 - `db_query_range()`
 - `db_query_temporary()`
 - DDL (CREATE TABLE, etc.)
 - INSERT/UPDATE/DELETE

Hard abstraction

- Field level
 - int vs. float vs. 'string'
 - db_encode_blob() (PostgreSQL only)
 - LOB handling (Oracle/DB2 only)
 - String parsing == evil
- Syntax level
 - Date handling
 - String parsing

Sad truth

There is no such thing as
standard SQL

SQL Abstraction

- More databases => less SQL
- Data abstraction?
- Feature emulation?
 - Clustering/replication
 - Transactions

Other questions

- Exceptions?
 - Most stable PDO error mechanism
 - Expose?
 - Trap?

Discussion

- What do we want to support?
- How much data abstraction do we want?
- What additional features do we want?
- Exposed exceptions, yay or nay?