

# 上海众森计算机科技有限公司



## SQL 规范

--

提交至:
所有者:
作者: 上海众森计算机科技有限公司 <a href="http://www.zerustech.com">www.zerustech.com</a>
编辑者: Michael Lee
版本: 1.2
日期: 2007/09/11

## 免责声明

本文件只用于信息沟通目的，不能作为上海众森计算机科技有限公司对于商品，服务的质量承诺书。上海众森计算机科技有限公司对本文件的一切内容拥有最终解释权。本文件内容如有任何改动，将不做特别通知。

## 文件修改履历

版本	修改日	修改人	注释
1.0	2007/04/09	Michael Lee	初稿
1.1	2007/05/14	Michael Lee	Fixing formatting issues. Change 'default' to 'text body'
1.2	2007/07/31	Michael Lee	Modifying copyright statement

## 目录

1 SQL.....	4
2 SQL 标准.....	4
3 语法.....	4
4 标识符.....	5
5 可读性.....	5
6 As 关键字.....	6
7 文本处理.....	6
8 索引.....	7
9 数据库细节.....	7
10 附录.....	8
10.1 词汇表.....	8
10.2 引用.....	9

# 1 SQL

由于 eZ publish 的数据库是独立的，所以需要有一个 SQLs，以便在很多数据库上运行。为了确保更大的兼容性，在 eZ publish 里面自定义的 SQL 应该遵循下面的规则和指导方针。

## 2 SQL 标准

一般来说，可以遵循 SQL92 的标准。因为在很多情况下它已经很好的支持了各式各样的数据库。这篇文档将会解释在什么情况下标准不会被遵循。

## 3 语法

SQL 语句都是由大写字母组成的。下面的例子显示了规范的 SQL 语句。

```
SELECT * FROM
    ezuser
WHERE
    id = '$user_id'
HAVING
    age > '42'
LIMIT 0,20

SELECT table_a,table_b AS c FROM
    ezcontentobject,
    ezcontentobject_attribute,
    ezcontentobject_version
WHERE
    ezcontentobject.id = ezcontentobject_version.contentobject_id AND
    ezcontentobject.id = '42'
LIMIT 0,20
```

## 4 标识符

标识符包括表，列，触发器和索引。

标识符应该遵循下面的规则：

- 所有的标识符都应该使用英文并且拼写正确。
- 多个单词应该用下划线来连接，类似 `contentobject` 的单词应该连在一起。
- 最长的英文单词不能超过 30 个字符，以确保该单词在数据库中有效。
- 如果一个单词实在很长(从单词长度来考虑)尝试用一个很好记忆的单词缩写来代替，建议使用较短的同义词。
- `eZ publish` 的核心的函数都应该在标识符前面加上 `ez` 前缀,另外的系统则在标识符前面加相应的前缀。

例子：

```
Tables:
ezuser
ezcontentobject_version
ezcontentclass_attribute

Columns:
id
first_name
last_name
login
```

## 5 可读性

尽量使得其他的开发者能够知道你的 SQL 语句的作用。

一些有效的指导方针：

- 缩进 SQL 语句，该行语句必须和上一行语句对齐。如果一个语句是在分隔线处开始，要缩进 2 空格。将 `SELECT`, `FROM` and `WHERE` 等关键字放到行首并重新设置语法缩进。
- 在一行里面，元素的数量应该限定在一到两个。所有的元素在一行里面显示会使得代码的可读性变得很差。

- 标识符和操作符应该用空格分开。
- 保持 AND 和 OR 语句在行尾

#### AND 和 OR 的示例

```
id = 5 AND  
depth > 5  
  
id = 2 OR  
status = 1
```

## 6 As 关键字

在定义表的别名的时候不能出现 **AS**，因为有些数据库允许但是有些数据库并不允许。换句话说，你应该将：

```
SELECT t1.a, t2.b FROM table1 t1, table2 t2 WHERE [condition];
```

来替换

```
SELECT t1.a, t2.b FROM table1 AS t1, table2 AS t2 WHERE [condition];
```

## 7 文本处理

如果有一列定义的类型是文本类型，那么你在插入数据的时候必须小心。如果精确插入数据库中的数据大小超过 2000 比特，那么应该用有约束的语句(参阅 `eZDBInterface::bindValue()`)来代替 SQL 语句。

例如：

```
INSERT INTO emp (empno,ename,sal) VALUES (2365,'VERYVERYLONGNAME',2000);
```

你不能明确指定 **employee name** 的长度，你因该通过有约束的变量来指定 **name**(在这个例子中是用 `:ename`):

```
INSERT INTO emp (empno,ename,sal) VALUES (2365,:ename,2000);
```

另外 **WHERE** 语句不能用在列的类型是 **CLOB** 中：

```
SELECT * FROM ezcontentobject_attribute WHERE data_text = 'some string';
```

## 8 索引

在一个表中索引是提高操作速度的最好的方法，而且当建立一个新表或者在已经存在的表中加入字段的时候，必须要考虑索引。这里很难给索引定出标准的规则，这依赖于开发者。

Sometimes the only way to figure out index usage is to fill the tables with thousands or million of entries and then analysing the executed SQLs.

遵循的规则：

- 在类型是 **CLOB** 的列中不要创建索引，一些数据库不能处理这些。
- 键的长度不能超过 **3218** 个字符。为了增加长度，在一些数据库中设置了字符类型是 **UTF-8**，所以在安装的时候要测试键的长度。
- 在表里面不要创建太多的索引，太多的索引会使得插入和更新都很慢。最好的查询方法是关联表，再根据索引来查询

## 9 数据库细节

这个对于在一个数据库上执行 **SQL** 语句很有帮助。这样可以提高速度或者简化 **SQL** 语句。

比如说你在一个表里面插入了很多元素，在 **MYSQL** 里面也许是用一条语句来执行，但是在其他的数据库里面并不支持。

我们应该在 **MYSQL** 里面做最佳的优化：

```
$db =& eZDB::instance();
if ( $db->databaseName() == 'mysql' )
{
    $db->query( "INSERT INTO ..." ); // Do a multi-insert query
}
else
{
    foreach ( $list as $item )
    {
        $db->query( "INSERT INTO ..." ); // Do a normal SQL insert
    }
}
```

```
}
```

下面的数据库细节必须考虑

左/右连接

Oracle8 不支持左连接和右连接。这意味这你必须重新写 SQL 语句或者专门为 Oracle 写一套 SQL 语句。

左右连接的处理，在连接列的后面放置一个“+”。

例如

```
SELECT DISTINCT language_code, name FROM ezcontentobject_attribute LEFT JOIN
ezcontent_translation ON
locale = language_code
WHERE contentobject_id = '1' AND version = '1'
ORDER BY name
```

必须写成:

```
SELECT DISTINCT language_code, name FROM ezcontentobject_attribute, ezcontent_translation
WHERE contentobject_id = '58' AND version = '1' AND
    locale(+) = language_code
ORDER BY name
```

## 10 附录

### 10.1 词汇表

词汇	解释

附录 1: 词汇表



## 10.2 引用

引用	注释
上海众森计算机科技有限公司	<a href="http://www.zerustech.com">www.zerustech.com</a>

附录 2: 引用