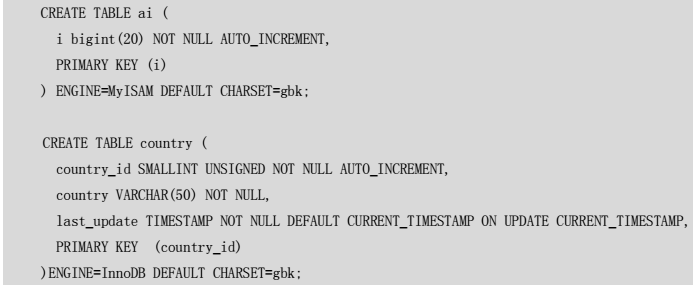
## 一MySql存储引擎

### 1.1 什么是存储

插件式存储引擎是mysql最重要的特性之一。用户可以根据需要选择如何存储和索引数据。是否使用事务等。Mysql默认支持多种存储引擎，以适用不同领域的数据需求。用户可以使用不同的存储引擎来提高应用的效率。

### 1.2 存储引擎变更



在创建表的时候指定存储引擎和表的字符集编码，当然如果不指定存储引擎，数据库会选用默认的存储引擎的方式进行存储。

alter table ai engine = innodb;

通过如上所示的sql可以改变表的存储引擎。

### 1.3 存储引擎的比较



### 1.4 MyISAM 存储引擎

MyISAM 不支持事务、也不支持外键，其优势是访问的速度快，对事务完整性没有要求或者以 SELECT、INSERT 为主的应用基本上都可以使用这个引擎来创建表。

MyISAM 的表又支持 3 种不同的存储格式: 静态（固定长度）表,动态表,压缩表。

静态表是默认的存储格式。一般选用静态表，如果选用动态表的话需要定期的通过OPTIMIZE TABLE命令整理内存中的碎片。

#### 1.4.1 静态表优点

静态表中的字段都是非变长字段，这样每个记录都是

固定长度的，这种存储方式的优点是存储非常迅速，容易缓存，出现故障容易恢复；

#### 1.4.2 静态表缺点

占用的空间通常比动态表多。静态表的数据在存储的时候会按照列的宽度定义补足空格，但

是在应用访问的时候并不会得到这些空格，这些空格在返回给应用之前已经去掉。

### 1.5 InnoDB

InnoDB 存储引擎提供了具有提交、回滚和崩溃恢复能力的事务安全。但是对比 MyISAM

的存储引擎，InnoDB 写的处理效率差一些并且会占用更多的磁盘空间以保留数据和索引。

在使用InnoDB引擎的时候可以设置自增长，并且通过“ALTER TABLE \*\*\* AUTO\_INCREMENT = n;对数据库自增长的初始值，进行调整。

对于 InnoDB 表，自动增长列必须是索引。如果是组合索引，也必须是组合索引的第一

列，但是对于 MyISAM 表，自动增长列可以是组合索引的其他列，这样插入记录后，自动增长列是按照组合索引的前面几列进行排序后递增的。

MySQL 支持外键的存储引擎只有 InnoDB，在创建外键的时候，要求父表必须有对应的

索引，子表在创建外键的时候也会自动创建对应的索引。

#### 1.5.1 创建外键

CREATE TABLE city (

city\_id SMALLINT UNSIGNED NOT NULL AUTO\_INCREMENT,

city VARCHAR(50) NOT NULL,

country\_id SMALLINT UNSIGNED NOT NULL,

last\_update TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (city\_id),

KEY idx\_fk\_country\_id (country\_id),

CONSTRAINT `fk\_city\_country` FOREIGN KEY (country\_id) REFERENCES country (country\_id) ON

DELETE RESTRICT ON UPDATE CASCADE

)ENGINE=INNODB DEFAULT CHARSET=utf8;

#### 1.5.2 外键约束

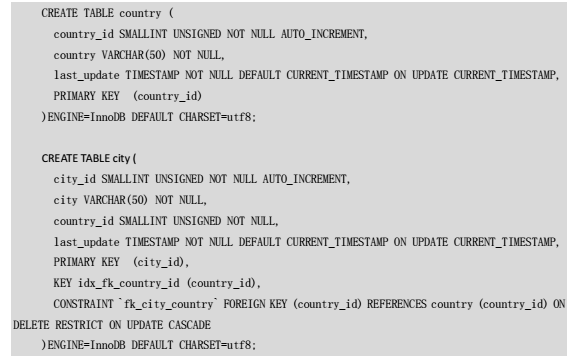
RESTRICT: 限制字表有关联记录的情况下父表不能更新，主要体现在删除上。

CASCADE: 表示父表在更新或者删除时，更新或者删除子表对应记录。

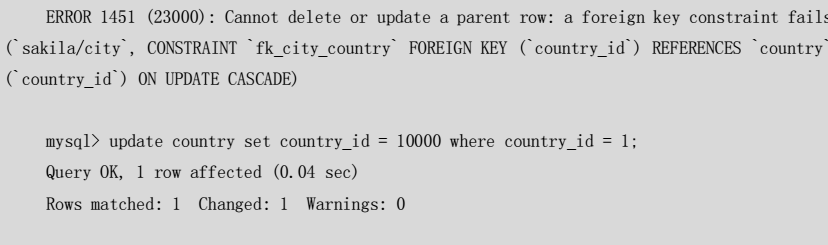
SET NULL: 表示父表在更新或者删除的时候，子表的对应字段被 SET NULL。

NO ACTION:其中 RESTRICT 和 NO ACTION 相同,限制字表有关联记录的情况下父表不能更新，主要体现在删除上。

例如 创建表



这个主表是country，副表示city。



在执行删除操作的时候受到了相应的约束。

### 1.6 选择存储引擎

MyISAM：默认的 MySQL 插件式存储引擎。如果应用是以读操作和插入操作为主只有很少的更新和删除操作，并且对事务的完整性、并发性要求不是很高，那么选择这个存储引擎是非常适合的。MyISAM 是在 Web、数据仓储和其他应用环境下最常使用的存储引擎

之一。

InnoDB：用于事务处理应用程序，支持外键。如果应用对事务的完整性有比较高的要求，在并发条件下要求数据的一致性，数据操作除了插入和查询以外，还包括很多的更新、删除操作，那么 InnoDB 存储引擎应该是比较合适的选择。InnoDB 存储引擎除了有效地降低由于删除和更新导致的锁定，还可以确保事务的完整提交（Commit）和回滚（Rollback），

对于类似计费系统或者财务系统等对数据准确性要求比较高的系统，InnoDB 都是合适的选

择。

MEMORY：将所有数据保存在 RAM 中，在需要快速定位记录和其他类似数据的环境下，可提供极快的访问。MEMORY 的缺陷是对表的大小有限制，太大的表无法 CACHE 在内存中，其次是要确保表的数据可以恢复，数据库异常终止后表中的数据是可以恢复的。MEMORY 表通常用于更新不太频繁的小表，用以快速得到访问结果。

MERGE：用于将一系列等同的 MyISAM 表以逻辑方式组合在一起，并作为一个对象

引用它们。MERGE 表的优点在于可以突破对单个 MyISAM 表大小的限制，并且通过将不同

的表分布在多个磁盘上，可以有效地改善MERGE表的访问效率。这对于诸如数据仓储等VLDB

环境十分适合