

# 试讲内容

## 导入

## 试讲稿脚本：MySQL数据操作管理

### 导入部分：

#### 讲师：

“上课，同学们好！”

“在开始上课前，老师请同学们想象这样一个场景：假设你正在运营一家在线书店，每天都有许多书籍被添加到库存中，客户下单购买，书籍售罄需要下架，甚至有时候需要调整书籍的价格或描述来吸引顾客。那请同学们思考一下，可以采取哪些方式来管理这些书籍信息呢？”

“老师给同学们3分钟的讨论时间，待会儿呀，咱们选派一个代表来回答这个问题。”

（3分钟后）

“好，时间到。我看A组的同学讨论得非常的激烈，那就请A组你们来说一下。”

（A组代表回答）

“嗯，很好！请坐。A组的同学说，可以选择使用基础的电子表格来记录书籍的信息并手动进行管理。嗯，A组提供的方法确实能够非常简单方便地实现书籍信息的管理。”

“那么B组呢？你们怎么看？”

（B组代表回答）

“嗯，很好，请坐。B组的同学说，随着业务的增长，可以将数据迁移到MySQL这样的关系数据库系统中，利用数据库提供的操作指令，实现数据的高效管理。”

“哎，非常好！看来B的同学学习得非常深刻。正如B组同学所说，我们可以通过使用像MySQL这样的关系数据库实现图书录入、下架以及信息更改操作，从而实现书籍信息的高效管理。那么同学们，你们知道MySQL是如何实现数据操作的吗？数据操作的方法又有哪些呢？那今天，就让同学们跟随老师的步伐，一起进入到今天关系数据库的操作方法的学习当中吧。”

### 板书：

关系数据库的操作方法

- 1、插入
- 2、更新
- 3、删除

讲师：

“我们先从插入数据开始。”

## 1. 插入数据

讲师：

“插入数据是向数据库中添加新记录的方式。MySQL提供了多种插入方式，让我们一一来学习。”

板书并解释：

```
不指定字段：INSERT INTO 表名 VALUES(V1, V2, ..., Vn)
```

讲师：“这种方式直接插入一整行数据，但必须确保值的数量和顺序与表中的列完全匹配。例如：”

```
INSERT INTO books VALUES(1, 'The Great Gatsby', 'F. Scott Fitzgerald',  
10.99, 100);
```

“在这个例子中，我们向表books插入了一条新记录，包含了书籍的ID、标题、作者、价格和库存量。这种方式要求所有字段的值都按照顺序提供。”

互动1：

讲师：“在这个例子中，我们直接指定了每个字段的值。如果你们要插入一本新的书籍记录，你们会怎么做？”

（等待学生回答，鼓励学生参与讨论，可以提供帮助和提示。）

讲师：“很好，比如你们可以插入一本新书《To Kill a Mockingbird》。有没有同学愿意尝试编写这条插入语句？”

（学生尝试回答。）

学生：“INSERT INTO books VALUES(2, 'To Kill a Mockingbird', 'Harper Lee', 12.99, 50);”

讲师：“非常好！这条语句向books表中插入了一本新书《To Kill a Mockingbird》，并指定了它的ID、标题、作者、价格和库存量。不过你们有没有想过，如果我们的表结构发生了变化，会怎样？这种方式还适用吗？”

（引导学生思考，帮助他们理解插入方式的灵活性和局限性。）

板书并解释：

指定字段: INSERT INTO 表名(C1, C2, ..., Cm) VALUES(V1, V2, ..., Vm)

讲师: “另一种方式是指定字段, 这样我们只须提供需要插入的字段和对应的值。例如: ”

```
INSERT INTO books (book_id, title, author) VALUES (3, '1984', 'George Orwell');
```

“在这个例子中, 我们只插入了书籍的ID、标题和作者, 没有指定价格和库存。这种方式更加灵活, 因为我们只须提供必要的字段。”

### 互动2:

讲师: “这种方式相对灵活, 我们只须提供需要插入的字段和对应的值。但有没有想过, 如果我们忘记指定一个字段会怎样?”

(引导学生思考, 让他们认识到指定字段方式的灵活性和需要注意的地方。)

学生: “可能会报错, 或者插入不完整的数据。”

讲师: “没错, 如果表中的某些字段是必填的而我们没有提供值, 就会导致错误或者插入的数据不完整。”

### 板书并解释:

多条插入: INSERT INTO 表名[(字段名列表)] VALUES(取值列表1), (取值列表2), (取值列表n)

讲师: “最后, 我们可以一次性插入多条记录, 这样可以提高效率。例如: ”

```
INSERT INTO books (book_id, title, author, price, stock)
VALUES
(4, 'Pride and Prejudice', 'Jane Austen', 8.99, 150),
(5, 'The Catcher in the Rye', 'J.D. Salinger', 9.99, 120);
```

“这条语句一次性插入了两本书《Pride and Prejudice》和《The Catcher in the Rye》, 每本书都指定了ID、标题、作者、价格和库存。”

### 互动3:

讲师: “一次性插入多条记录的方式, 是不是感觉更高效呢? 有没有同学尝试过这种方式?”

(鼓励学生分享经验或观点, 并引导他们思考这种方式在实际应用中的优势。)

学生: “这样可以节省时间, 减少插入操作的次数。”

讲师: “对, 这种方式在需要批量插入数据时非常高效, 特别是在处理大量数据的时候。”

## 2. 更新数据

讲师：

“接下来，我们看看如何更新数据。”

板书并解释：

```
UPDATE 表名  
SET C1 = V1, C2 = V2, ..., Cn = Vn  
[WHERE 条件表达式]
```

讲师：“更新操作可以修改表中的现有记录。这里我们使用 **SET** 来指定要更新的列和值，**WHERE** 条件用于限定要更新的记录。例如： ”

```
UPDATE books  
SET price = 12.99, stock = 80  
WHERE book_id = 1;
```

“这条语句更新了 **book\_id** 为1的记录，将价格改为12.99，将库存改为80。”

互动4：

讲师：“大家看到这条语句了吗？谁能告诉我，这里发生了什么？”

（引导学生回答，让他们解释语句的含义和作用。）

学生：“这条语句更新了书籍《The Great Gatsby》的价格和库存。”

讲师：“没错，这条语句对 **book\_id** 为1的记录进行了更新，将 **price** 和 **stock** 字段的值进行了修改。”

## 3. 删除数据

讲师：

“最后，我们来学习删除数据。”

板书并解释：

```
DELETE FROM 表名 [WHERE 条件表达式]
```

讲师：“删除操作用于移除表中的记录，**WHERE** 条件用于限定要删除的记录。如果不加 **WHERE** 条件，会删除表中的所有记录。我们来看一个例子： ”

```
DELETE FROM books  
WHERE book_id = 4;
```

“这条语句删除了 `book_id` 为4的记录，即书籍《Pride and Prejudice》。”

### 互动5:

**讲师:** “谁能告诉我，这条语句会删除哪条记录？”

(引导学生回答，让他们理解语句的含义和作用。)

**学生:** “这条语句会删除书籍《Pride and Prejudice》的记录。”

**讲师:** “对，这条语句会删除 `book_id` 为4的记录。如果不加 `WHERE` 条件，会发生什么呢？”

(等待学生回答。)

**学生:** “会删除表中的所有记录。”

**讲师:** “对，会删除表中的所有记录。所以删除操作时最需要注意的就是小心使用 `WHERE` 条件，避免误删除记录。”

---

### 总结:

“今天我们学习了如何在MySQL中进行数据的插入、更新和删除操作。希望大家通过这些例子能够更好地理解和掌握这些基本操作。接下来，我们将进行一些实际操作练习，大家可以按照我们今天讲的内容来尝试。”

“谢谢大家！”

---