

创建表:

```
CREATE TABLE table_name  
(  
    field1 datatype,  
    field2 datatype,  
    field3 datatype  
)character set 字符集 collate 校对规则 engine 存储引擎  
field: 指定列名 datatype: 指定列类型(字段类型)  
character set: 如不指定则为所在数据库字符集  
collate: 如不指定则为所在数据库校对规则  
engine: 引擎(这个涉及内容较多, 后面单独讲解)
```

文本类型:

CHAR(size): 固定长度字符串 最大 255 字符

VARCHAR(size): 0-65535 字节

可变长度字符串 最大 65532 字节 【utf8 编码最大 21844 字符 1-3 个字节
用于记录大小】

使用细节:

1. char(4) 和 varchar(4) 这个 4 表示的是字符, 而不是字节, 不区分字符是汉字还是字母
2. 定长情况使用 char(), 长度不确定时, 使用 varchar()。
3. 查询速度: char>varchar

插入 insert:

细节:

1. 插入的数据应与字段的数据类型相同。
2. 数据的长度应在列的规定范围内
3. 在 values 中列出的数据位置必须与被加入的列的排列位置相对应。
4. 字符和日期型数据应包含在单引号中。
5. 列可以插入空值[前提是该字段允许为空], insert into table value(null)
6. insert into tab_name (列名..) values (), (), () 形式添加多条记录
7. 如果是给表中的所有字段添加数据, 可以不写前面的字段名称
8. 默认值的使用, 当不给某个字段值时, 如果有默认值就会添加默认值, 否则报错

a. 如果某个列 没有指定 not null ,那么当添加数据时, 没有给定值, 则会默认给 null

b. 如果我们希望指定某个列的默认值, 可以在创建表时指定更新 update:

1. UPDATE语法可以用新值更新原有表行中的各列。
2. SET子句指示要修改哪些列和要给予哪些值。
3. WHERE子句指定应更新哪些行。如没有WHERE子句, 则更新所有的行(记录), 因此老师提醒一定小心。
4. 如果需要修改多个字段, 可以通过 set 字段1=值1, 字段2=值2....

删除 delete:

1. 不使用 where 子句, 删除表中的所有数据
2. delete 不能够删除某一列的值
3. delete 语句仅删除记录, 不删除表格本身。

查询 selete:

1. selete 指定查询哪些列的数据
2. distinct 可以去除重复数据。

Where 子句:

比较运算符	> < <= >= = <> !=	大于、小于、大于(小于)等于、不等于
	BETWEEN ...AND...	显示在某一区间的值
	IN(set)	显示在in列表中的值, 例: in(100,200)
	LIKE '张pattern' NOT LIKE "	模糊查询 模糊查询
	IS NULL	判断是否为空
逻辑运算符	and	多个条件同时成立
	or	多个条件任一成立
	not	不成立, 例: where not(salary>100);

排序 order by:

1. 指定排序的列
2. asc 表示升序 (默认), desc 表示降序
3. 该子句位于 selete 语句末尾。

流程控制函数:

1. IF(expr1, expr2, expr3) 如果 expr1 为 True ,则返回 expr2 否则返回 expr3

2. IFNULL(expr1, expr2) 如果 expr1 不为空 NULL, 则返回 expr1, 否则返回 expr2

3. SELECT CASE WHEN expr1 THEN expr2 WHEN expr3 THEN expr4 ELSE expr5 END; [类似多重分支.]

如果 expr1 为 TRUE, 则返回 expr2, 如果 expr2 为 t, 返回 expr4, 否则返回 expr5

查询加强:

1. Like 操作符:

%: 表示 0 到多个字符。

_ : 表示单个字符。

2. 在 mysql 中, 日期类型可以直接比较, 需要注意格式。

3. 分页查询:

```
-- 推导一个公式
SELECT * FROM emp
ORDER BY empno
LIMIT 每页显示记录数 * (第几页-1), 每页显示记录数
```

多表查询:

1. 多表查询的条件不能少于 表的个数-1, 否则会出现笛卡尔集

2. 自连接的特点

a. 把同一张表当做两张表使用

b. 需要给表取别名 表名 表别名

c. 列名不明确, 可以指定列的别名 列名 as 列的别名

3. 子查询:

子查询是指嵌入在其它 sql 语句中的 select 语句, 也叫嵌套查询

单行子查询是指只返回一行数据的子查询语句。

多行子查询是指返回多行数据的子查询语句, 与 in 配合使用。

多列子查询: 返回多个列数据的子查询语句。

表复制:

- (1) 先创建一张临时表 `my_tmp` , 该表的结构和 `my_tab02` 一样
- (2) 把 `my_tmp` 的记录 通过 `distinct` 关键字 处理后 把记录复制到 `my_tmp`
- (3) 清除掉 `my_tab02` 记录
- (4) 把 `my_tmp` 表的记录复制到 `my_tab02`
- (5) drop 掉 临时表 `my_tmp`

合并查询: `Union all` 与 `union`

`Union all`: 不取消重复行。

`Union`: 会取消重复行。

约束:

1. 主键 `primary key`

- a. `primary key` 不能重复而且不能为 `null`。
- b. 一张表最多只能有一个主键, 但可以是复合主键
- c. 使用 `desc` 表名, 可以看到 `primary key` 的情况

2. 非空 `not null`

插入数据, 必须为该列提供数据

3. 唯一 `unique`

该列值不可重复

- a. 如果没有指定 `not null` , 则 `unique` 字段可以有多个 `null`
- b. 如果一个列(字段), 是 `unique not null` 使用效果类似 `primary key`
- c. 一张表可以有多个 `unique` 字段

4. 外键 `foreign key`

- a. 定义在从表上, 主表应该有主键或者 `unique` 约束
- b. 指向的字段要为主键或者唯一
- c. 表格的 `engine` 应该为 `innodb`
- d. 外键与主键字段的类型应一致, 长度可不同
- e. 外键中的值必须在主键之中出现过或者为 `null`
- f. 建立主外键关系, 不能随便删除

5. 检查 `check`

索引:

1. 主键索引, 主键自动的为主索引 (类型Primary key)
2. 唯一索引 (UNIQUE)
3. 普通索引 (INDEX)
4. 全文索引 (FULLTEXT) [适用于MyISAM]
一般开发, 不使用mysql自带的全文索引, 而是使用 : 全文搜索 Solr 和 ElasticSearch (ES)

事务:

事务用于保证数据的一致性,它由**一组相关的dml语句组成**,该组的dml语句要么全部成功, 要么全部失败。如:转账就要用事务来处理, 用以保证数据的一致性。

start transaction:开始事务

savepoint 保存点名: 设置保存点

rollback to 保存点名: 回退事务

rollback: 回退全部事务

commit: 提交

细节:

1. 如果不开始事务, 默认情况下, dml 操作是自动提交的, 不能回滚
2. 如果开始一个事务, 你没有创建保存点. 你可以执行 rollback, 默认就是回退到你事务开始的状态
3. 你也可以在这个事务中(还没有提交时), 创建多个保存点。
4. 你可以在事务没有提交前, 选择回退到哪个保存点
5. InnoDB 存储引擎支持事务。

事务隔离:

概念: Mysql 隔离级别定义了事务与事务之间的隔离程度。

Mysql 隔离级别 (4种)	脏读	不可重复读	幻读	加锁读
读未提交 (Read uncommitted)	V	V	V	不加锁
读已提交 (Read committed)	x	V	V	不加锁
可重复读 (Repeatable read)	x	x	X	不加锁
可串行化 (Serializable) [演示重开客户端]	x	x	x	加锁

说明: V 可能出现 x 不会出现