

MySQL_Advance

普通查询

整体结构:

SELECT ... 聚合函数 FROM table_name
WHERE ...
GROUP BY ...
HAVING ...
ORDER BY ...
LIMIT ...;

-> 6
-> 1
-> 2
-> 3
-> 4
-> 5

执行顺序

聚合函数

AVG(feild_name) 该字段的平均值
MAX(feild_name) 该字段的最大值
MIN(feild_name) 该字段的最小值
SUM(feild_name) 该字段所有记录的和
COUNT(feild_name) 统计该字段记录的个数 字段为 NULL 不能统计

GROUP BY

给查询的结果进行分组
先分组再聚会
GROUP BY 的字段名必须出现在 SELECT 的后面
SELECT 后面的字段名没有在 GROUP BY 后面出现, 必须做聚合处理

HAVING

对分组聚合后的结果进行进一步筛选
WHERE 只能操作表中实际存在的字段
HAVING 操作的是聚合函数生成的显示列

区别

DISTINCT

去重 SELECT DISTINCT field_name,... FROM table_name
DISTINCT 和 FROM 之间所有字段都相同才会去重
DISTINCT 不能对任何字段做聚合处理

与运算符一起使用

select field_name1,field_name2*2 from table_name;
update table_name set field_name=field_name*2;
...

高级查询

嵌套查询 SELECT...FROM table_name WHERE condition (SELECT...);

多表查询 SELECT 字段名列表 FROM 表名列表 WHERE 条件;

连接查询

SELECT field_name FROM table_name1
INNER JOIN table_name2 ON CONDITION
INNER JOIN table_name3 ON CONDITION

INNER 表示内连接, 结果同多表查询,显示匹配到的记录
LEFT 表示内连接, 以左边为主显示查询结果
RIGHT 表示内连接, 以右边为主显示查询结果

索引

普通(MUL)、唯一(UNI)索引

创建

CREATE TABLE table_name(
...
INDEX(field_name),
...
)

创建表时

CREATE INDEX index_name ON table_name(field_name);

已有表时

查询

DESC table_name; -> KEY
SHOW INDEX FROM table_name\G;

Non_Unique:1 -> index
Non_Unique:0 -> unique

删除

DROP INDEX index_name on table_name;

主键(PRI)、自增长(AUTO_INCREMENT)

创建主键

CREATE TABLE table_name(
id INT PRIMARY KEY AUTO_INCREMENT,
...
)CHARSET=utf8,AUTO_INCREMENT=1000; # 自增长起始值

可以类似普通索引 创建 PRIMARY KEY(id) (不常用)
创建表时

ALTER TABLE table_name ADD PRIMARY KEY(id);

已有表时 (不常用)

已有表操作自增长属性

ALTER TABLE table_name MODIFY field_name INT AUTO_INCREMENT;

已有表添加自增长属性

ALTER TABLE table_name AUTO_INCREMENT=2000;

已有表重新指定起始值

删除

删除自增长属性 (modify)
删除主键索引: ALTER TABLE table_name DROP PRIMARY KEY;

删除组件必须先删除自增长, 在删除主键, 否则删除失败

外键

创建

FOREIGN KEY (参考字段名)
REFERENCES 主表 (被参考字段名)
ON DELETE 级联动作
ON UPDATE 级联动作

CREATE TABLE slave_tname(
stu_id INT,
name VARCHAR(20),
money DECIMAL(10,2),
FOREIGN KEY (stu_id) REFERENCES master_tname(id)
ON DELETE CASCADE ON UPDATE CASCADE
)CHARSET=utf8;

在从表创建外键

级联动作

CASCADE 数据级联删除、更新(参考字段)
RESTRICT(默认) 从表有相关联记录,不允许主表操作
SET NULL 主表删除、更新,从表相关联记录字段值为NULL

删除

ALTER TABLE table_name DROP FOREIGN KEY foreign_name;

foreign_name 用查询创建表语句查询

添加

ALTER TABLE slave ADD
FOREIGN KEY (stu_id) REFERENCES MASTER(id)
ON DELETE SET NULL
ON UPDATE SET NULL;

数据导入/导出

导入

LOAD DATA INFILE "文件名"
INTO TABLE 表名
FIELDS TERMINATED BY "分隔符"
LINES TERMINATED BY "\n"

步骤

将 表文件 放到数据库搜索路径中 (Ubuntu系统独有) /var/lib/mysql-files/
在数据库中创建对应的表
执行数据导入语句

导出

mysql> source sql文件绝对路径;
SELECT ... FROM table_name
INTO OUTFILE "file_name"
FIELDS TERMINATED BY "separator"
LINES TERMINATED BY "separator";

将 表文件 放到数据库搜索路径中 (Ubuntu系统独有) /var/lib/mysql-files/

账户管理

开启 MySQL 远程连接

打开 配置文件 /etc/mysql/mysql.conf.d/mysql.d.cnf

更改配置

注释: bind-address = 127.0.0.1
添加: character_set_serive=utf8

重启: /etc/init.d/mysql restart

添加授权用户

用 root 用户登录 mysql

grant 授权列表 on 库.表
to "用户名"@"%" identified by "密码"
with grand option;

授权列表: all privileges、select、insert ...
库.表: *.* 所有库所有表
% -> 任何地址
with grand option -> 继续授权

刷新权限 flush privileges;

ER模型

一对一 主外键关联 + 外键字段添加唯一索引

create table t1(
t1_id int primary key auto_increment
);
create table t2(
t2_id int unique,
foreign key (t2_id) references t1(t1_id)
);

一对多 主外键关联

create table middel(
id int primary key auto_increment,
t_id int,
c_id int,
foreign key (t_id)
references teacher(id),
foreign key (c_id)
references course(id)
)charset=utf8;

多对多 建立中间表, 中间表非主键, 不唯一字段外键关联, 这两个表

其他操作

异常退出操作: Ctrl C + ENTER

SHOW VARIABLES LIKE '%character%';

查看 mysql 所有默认变量, 可使用模糊查询

时间监测

临时开启 运行时间监测 set profiling=1;
查询 运行时间 show profiles;

修改已有库中的字符编码

alter database db_name charser utf8;

表的复制

CREATE TABLE 表名 SELECT 查询命令;
CREATE TABLE 表名 SELECT 查询命令 WHERE FALSE;

复制表结构

复制表的时候不会把原有表的键属性复制过来