## 一：基本命令：

#### 查看当前服务下的所有数据库

SHOW DATABASES;

#### 创建数据库

CREATE DATABASE mydb;

#### 进入数据库

USE mydb;

#### 查看当前库中的表

SHOW TABLES;

#### 创建数据表

CREATE TABLE classinfo(  
 classId int AUTO\_INCREMENT PRIMARY KEY,  
 className varchar(4),  
 openDate date  
);

#### 查看表结构

DESC classinfo;// description描述

**数据操作：**

#### 新增一行记录

INSERT INTO classinfo(classId, className, Opendate)  
 VALUE (1,'KB06','2020-1-18');

#### 新增多行记录

INSERT INTO classinfo(classId, className, Opendate)   
 VALUE (6,'KB09','2020-4-18'),  
 (9,'KB07','2020-2-18'),  
 (10,'KB08','2020-3-18');

#### 新增多行记录----从一张表中的数据搬到另一张表

INSERT INTO classinfo2(classId, className, Opendate)  
SELECT  
classId, className, Opendate  
FROM classinfo;

#### 新增多行记录----从一张表中的结构和数据搬到新表

CREATE TABLE classinfo3(  
 SELECT classId, className, Opendate FROM classinfo  
);

#### 删除数据记录

DELETE FROM classinfo3 WHERE classId=3;

#### 修改数据记录

#### 修改列：

UPDATE classinfo3 SET className='TB08',openDate='2020-4-10'  
 WHERE classId=2;

#### 简单查询语句

SELECT \* FROM classinfo;

**对象操作：**

#### 删除数据库

DROP DATABASE mydb;

#### 删除数据表

DROP TABLE classinfo3;

#### 删除数据----一次性删除表中所有数据，无法恢复--不可以带条件

TRUNCATE TABLE classinfo2;

**创建带约束的表**

CREATE TABLE student(// 从表  
 stuId INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 stuName VARCHAR(4) NOT NULL,  
 gender CHAR(1) NOT NULL DEFAULT '男',  
 email VARCHAR(50) NOT NULL UNIQUE KEY,  
 // 这张表是外键，在classinfo是主键，这里外键引用主键  
 //classId INT NOT NULL REFERENCES classinfo(classId)  
 // fk\_主表名\_从表名\_外键字段名  
 classId INT NOT NULL,  
 CONSTRAINT fk\_class\_student\_classId FOREIGN KEY(classId) REFERENCES classinfo(classId)  
 // 被引用的主键的表是主表  
)CHARSET UTF8;

插入时：1.自增列不管它，2.有默认值的字段也可以不管它，除非要改，3.允许为空也可以不管

INSERT INTO student(stuName, email, classId)   
 VALUE('李斯','123@ww.com','1');  
   
INSERT INTO student(stuName,gender,email, classId)   
 VALUE('李斯','女','123@sww.com','7');  
INSERT INTO student(gender,email, classId)   
 VALUE('女','123@ww.com','6');

创建基本语法：

CREATE TABLE TABLE\_NAME(  
 FIELD\_NAME DATA\_TYPE(n) SPECIAL\_CONSTRAINT,  
 ...  
);

新增数据基本语法：

INSERT INTO TABLE\_NAME(FIELD\_NAME,...) VALUE (VALUE,...);

删除数据基本语法：

DELETE FROM TABLE\_NAME WHERE CONDITION;

修改数据基本语法：

UPDATE TABLE\_NAME SET FIELD\_NAME1=VALUE1, FIELD\_NAME2=VALUE2,... WHERE CONDITION;

## 二：数据类型：

|  |  |  |  |
| --- | --- | --- | --- |
|  | java | js | mysql |
| 字符 | char | string | char（n）-n为允许的最大长度 |
| 字符串 | String |  | varchar(n)-可变长度 |
| 字符串 |  |  | text-长文本 |
| 数值 | byte | number | tinyint |
|  | short |  | smallint |
|  | int |  | int |
|  | long |  | bigint |
| 小数 | float |  | decimal(m,n) |
|  | double |  | -m:最大长度(整体)-n:精度(小数) |
|  |  |  | money |
|  |  |  | numeric(m,n) |
| 布尔 | boolean | boolean | bit 0/1(取到java变成T/F) |
| 日期 | Date | Date | date年月日 |
|  |  |  | datetime年月日时分秒 |
|  |  |  | timestamp自动提供默认值 |

## 三：SPECIAL\_CONSTRAINT

#### 1.符号：默认有正负-------UNSIGNED:无符号（只有正数）

#### 2.整数：自增列，AUTO\_INCREMENT ，默认1~n

#### 3.非空：NOT NULL

#### 4.默认值：DEFAULT VALUE

#### 5.唯一键：UNIQUE KEY

#### 6.主键：PRIMARY KEY

**PS：关系型数据库，一般每张表都需要设置一个无意义的自增列作为主键，主键为不可以重复，不可以为null的字段。**

#### 7.外键：FOREIGN KEY

## 四：查询

#### 删除/修改/查询：CONDITION

1. 一般来说，删除和修改必须带条件，且条件是主键或唯一键
2. 查询条件非常丰富：
   * SELECT .... WHERE field\_name

|  |  |  |
| --- | --- | --- |
|  | * + java | * + mysql |
| * + 与 | * + && | * + and |
| * + 或 | * + || | * + or |
| * + 非 | * + ！ | * + not |

* + 筛选：SELECT ... WHERE field\_name IS [NOT] NULL
  + 区间：SELECT ... WHERE number\_field BETWEEN ... AND...
    - 注意:between(>=) and(<=)
  + 列举：SELECT ... WHERE field\_name IN（...）
  + 模糊查：SELECT ... WHERE field\_name LIKE '.....'
    - 通配符
      1. % ：通配任意长度任意内容
      2. \_： 统配一个长度任意内容

#### 查询

1. 基础查询标准语法

* SELECT  
   field1,....field2 [as] alias(别名) 主要用于额外字段的别名 (字段名)  
  FROM  
   table  
  WHERE  
   ... (字段里的具体数据)
* PS:字段列表为\*为所有字段

#### 1.简单条件查询

* + AND语句
* SELECT  
   \*  
  FROM  
   student  
  WHERE  
   gender='男'  
  AND  
   classId=2;
  + AND加关系运算符
* SELECT  
   \*  
  FROM  
   student  
  WHERE  
   gender='男'  
  AND  
   stuId>=2;

#### 2.排序查询

* ORDER BY main\_field ASC (升序) / DESC (降序), sub\_field ASC (升序) / DESC (降序)
  + **注意:**1.主字段排序完后相等的,按子字段排序
  + 2.不写升降序，默认升序

#### 3.分页查询

* 当前页码为pageNo,一页显示的数据量为pageSize
* limit (pageNo-1)\*pageSize, pageSize;

#### 4.分组查询

* SELECT  
  [main\_field, sub\_field....],聚合函数列表  
  FROM  
  stuscore  
  GROUP BY  
  main\_field, sub\_field...;
  + 注意：只有在group by之后出现，才可以出现在select之后的字段列表
  + 先根据主字段分组，再根据子字段分组
* **给每个subId分组显示**
* SELECT  
  subId, AVG(score) avgSorce  
  FROM  
  stuscore  
  GROUP BY  
  subId;

## 五：函数

#### 字符串函数

* 获取字符串的字节长度： length('...') 汉字三个字节
* 获取字符串长度： char\_length('...')
* 拼接字符串： concat(val1,...,valn)
* 转小写/大写： lower(string)upper(string)
* 左右截取字符串: left('...', 从做开始长度)，right('...', 长度)
* 从指定位置截指定长度: mid/substring(str,pos,len) {字符串,开始位置，长度}从1开始
* 去除字符串两端空格; trim(string)
* 字符串替换: replace(str, src, dest){字符串，源，目标}

#### 日期函数

* 获取指定日期 select curdate();
* 获取当前时间 select now();
* 获取unix时间戳 unix\_timestamp(),返回长整型
* 获取指定日期的年 year(date) -----date='2020-12-20'
* 获取指定日期的月 month(date)
* 获取指定日期的日 dayofmonth(date)
* 获取指定日期是星期几 dayofweek(date) 周日~周六：1~7
* 获取指定日期是该年的第多少天 dayofyear(date)
* 指定日期的指定部分加值后的日期 adddate(date,interval n format) ---format='day,month,year,second,minute....'
* 指定日期的指定部分减值后的日期 subdate(date,interval n format) ---format='day,month,year,second,minute....'
* 获取指定日期的星期索引 weekday(date) 周一~周日：0~6
* 获取两个日期的间隔天数差 datediff(dateBig, dateSmall) ；

#### 数学函数

* 求绝对值 abs(num);
* 向上取整 ceil(decimal);decimal='0.04...'
* 向下取整 floor(decimal);

#### 聚合函数

* 计数 count(1/\*/field\_name)
* // count( \* ）对行数目计数，count(字段名)对字段名不为空的计数
* 求和 sum(num)
* 均值 avg(num)
* 最大值 max(num)
* 最小值 min(num)

## 六：事务 : Transaction

#### 4个特征：

* **原子性：**（atomicity）
* **一致性：**（consistency）
* **隔离性：**(isolation)
* **持久性：**（durability）

#### 1.为了保证数据的一致性

#### 2.应用场景：多条增，删，改语句构成复合业务

#### 3.事务的保障

* commit 提交数据--->将操作从缓存中持久化（落盘）到磁盘文件
* rollback 回滚数据--->将本次事务中的操作结果从缓存中清除

SQL-92标准

* T-sql 增强型sql语言
* pl-sql 可编程sql语言
* pg-sql

时序数据库

图形数据库

NoSQL

redis

mongodb

hbase