SQL基础知识笔记

1. SQL基础知识笔记
    - 1.1 一、SQL的基本功能
        - 1.1.1 1. 数据定义（DDL）
        - 1.1.2 2. 数据操纵（DML）
        - 1.1.3 3. 数据控制（DCL）
    - 1.2 二、SQL查询的处理步骤
    - 1.3 三、数据的获取
        - 1.3.1 1. 对获取的行数进行限制
        - 1.3.2 2. JSON 列解析
        - 1.3.3 3. 对结果进行排序
    - 1.4 四、数据预处理
        - 1.4.1 1. 缺失值处理
        - 1.4.2 2. 重复值处理
        - 1.4.3 3. 数据类型转换
        - 1.4.4 4. 重命名
    - 1.5 五、数据运算
        - 1.5.1 1. 算术运算
        - 1.5.2 2. 比较运算
        - 1.5.3 3. 逻辑运算
        - 1.5.4 4. 常用数学函数
        - 1.5.5 5. 常用字符串函数
        - 1.5.6 6. 聚合函数
        - 1.5.7 7. 量词
    - 1.6 六、控制函数
        - 1.6.1 case when
    - 1.7 七、日期和时间函数
        - 1.7.1 1. 获取当前时刻的数据
        - 1.7.2 2. 日期和时间格式转换
        - 1.7.3 3. 日期和时间计算
    - 1.8 八、数据分组与数据透视表
    - 1.9 九、窗口函数
        - 1.9.1 1. 序列函数
        - 1.9.2 2. 参数
        - 1.9.3 3. grouping 运算符（与 group by 结合使用）
    - 1.10 十、多表连接
        - 1.10.1 1. 表的横向连接（合并）

1.10.2 2. 表连接的类型
1.10.3 3. 多张表连接
1.10.4 4. 标的纵向连接（追加）
1.10.5 5. 表的集合
1.11 十一、子查询
1.12 十二、视图
1.12.1 1. 创建视图
1.12.2 2. 删除视图
1.13 十三、SQL查询的执行顺序
1.13.1 1. 关键词
1.13.2 2. 执行顺序
1.14 十四、变量设置
1.15 十五、DDL
1.15.1 1. 创建表
1.15.2 2. 向表中插入数据
1.15.3 3. 修改表中的数据
1.15.4 4. 删除表
1.15.5 5. 复制表数据
1.15.6 6. 数据更新
1.15.7 7. 创建事务

一、SQL的基本功能

1. 数据定义（DDL）

- 关键词：create（创建）、drop（删除）、alter（修改）
- 对象：数据库、表、视图、索引

2. 数据操纵（DML）

- 关键词：select（查询）、insert（插入）、update（更新）、delete（删除）
- 对象：表

3. 数据控制（DCL）

- 关键词：grant（赋予权限）、revoke（取消权限）、commit（提交）、rollback（回滚）
- 对象：表和列
- SQL是一门大小写不敏感的语言，语句以英文半角分号（;）结尾，多行注释使用 /* …… */，单行注释使用 -- ……

- 练习网站：牛客网

二、SQL查询的处理步骤

- 查询分析
- 查询检查
- 查询优化
- 查询执行

三、数据的获取

1. 对获取的行数进行限制

- 使用 limit x 表示获取前 x 行数据
- 使用 limit x,y 表示获取第 x 行（不包括第 x 行）以后的 y 行数据
- 使用 limit x offset y 表示获取获取第 y 行（不包括第 y 行）以后的 x 行数据
- 使用 where 来指定具体的条件

2. JSON 列解析

- 使用 json_extract(JSON 格式的列名，具体 key 对应的 value 值)，注意 key 前面的 $. 不可少
- 使用 json_keys(JSON 格式的列名) 来获取 key，结果是列表 list 的形式

3. 对结果进行排序

- 使用 order by，默认是升序排序
- asc 升序排序，desc 降序排序
- 可以在 order by 子句中引用前面 select 查询中没有使用到的字段，但是如果在 select 中使用了 distinct 或 unique 关键字，就只允许使用在 select 查询中明确列出的字段
- 汉字排序规则，需要查阅相应数据库的文档，字符串类型的数据按照字典顺序进行排序，不能与数字的大小顺序混淆
- 排序键中包含 null 时，会在开头或结尾进行汇总

四、数据预处理

1. 缺失值处理

- 通过 where 进行过滤，直接过滤掉缺失值
- 使用 coalesce(null,null,……,null,value,……) 结合 case when 语句进行条件判断，对缺失值进行填充

2. 重复值处理

- 使用 distinct，在 select 中只等使用一次，且关键字必须放在第一位
- 对想要删除重复值的列进行 group by

3. 数据类型转换

- cast (value as type)
- convert (value,type)

类型	符号
浮点型	decimal
整型	signed
字符型	char
二进制	binary
日期	date
时间	time
日期时间	datetime

4. 重命名

- 使用 as 别名

五、数据运算

1. 算术运算

说明	符号
加	+
减	-
乘	*
除	/
整除	div
取余（模）	% 或 mod

- null 与任何值进行运算，结果都是 null

2. 比较运算

说明	符号
大于	>
小于	<
等于	=
大于等于	>=
小于等于	<=
不等于	<> 或 !=
(不) 介于	(not) between …… and……
(非) 空值	is (not) null
(不) 在集合中	(not) in
(不) 存在	(not) exists

- 比较结果正确则返回 1，不正确则返回 0
- (not) in 是无法选取出 null 数据的，常结合子查询使用
- (not) exsists 判断是否（不）满足某种条件的记录，常结合关联子查询使用，在关联子查询中常使用 select *

3. 逻辑运算

- and：与，多个条件均为真时结果才为真
- or：或，多个条件中只要有一个条件为真时结果就为真
- not：非，取反

4. 常用数学函数

- 求绝对值：abs(x)
- 求最小整数值：ceil(x)
- 求最大整数值：floor(x)
- 生成随机数：rand()，返回 0-1 范围内的一个随机浮点数
- 小数点位数调整：round(x,d)
- 正负判断：sign(x)，正数时返回 1 ，负数时返回 -1，0 时返回 0
- 空值判断：isnull(x)，null 时返回 1，否则返回 0

5. 常用字符串函数

- 字符串替换：replace(str,a,b)

- 字符串合并：concat(str1,str2,……,strn) 或 concat_ws(s,str1,str2,……,strn)
- 字符串截取：left(str,n) 或 right(str,n) 或 substring(str,m,n)
- 字符串匹配：like 结合 % 或 _ 或 [] 或 [^] 或 escape 's' 转义
- 字符串计数：基于字符计数 char_length(str) 或 基于字节计数 length(str)
- 英文字母 1 个字符 1 个字节，中文 1 个字符在 utf-8 下 3 个字节，在 gbk 下2 个字节
- 去除字符串空格：ltrim(str) 或 rtrim(str) 或 trim(str)
- 字符串重复：repeat(str,n)
- 转换为小写：lower(str)
- 转换为大写：upper(str)

6. 聚合函数

- count() 计数，用于对多个非缺失值进行计数，常用于查看表中某列有多少非空值，null 和空值不计数，空格会计数，去重计数可结合使用 distinct
- sum() 求和，null 和空值不计数，去重求和可结合使用 distinct
- avg() 秋平均值，null 和空值不计数，分母中也不包含 null 和空值
- max() 求最大值
- min() 求最小值
- var_pop() 求总体方差
- var_samp() 求样本方差
- std() 求总体标准差
- stddev_samp() 求样本标准差
- 聚合函数之间的运算需要写完整，不要直接使用别名

7. 量词

- all 操作符： < all，小于最小的； > all，大于最大的； = all，没有返回值
- any 操作符： < any，小于最大的； > any，大于最小的； = any，等于 in
- 量词用于比较子查询返回列表中的每一个值

六、控制函数

case when

- 形式一：

```
case 列名
    when 条件1 then 返回值1
    when 条件2 then 返回值2
    ……
    when 条件n then 返回值n
```

```
    else 返回默认值
end
```

- 形式二：

```
case
    when 列名满足条件1 then 返回值1
    when 列名满足条件2 then 返回值2
    ……
    when 列名满足条件n then 返回值n
    else 返回默认值
end
```

- 形式一的条件只能是具体的值，不能进行比较运算，形式二是支持比较运算的

七、日期和时间函数

- 日期是指年月日，时间是指时分秒

1. 获取当前时刻的数据

- now()
- curdate()
- date(now())
- year(now())
- month(now())
- day(now())
- curtime()
- time(now())
- hour(now())
- minute(now())
- second(now())
- weekofyear(now())
- dayofweek(now())
- quarter(now())

2. 日期和时间格式转换

- date_format(datetime,format)

主题	format	描述
年	%Y	4 位数字表示的年
月	%b	月份对应的英文缩写
月	%M	月份对应的英文全称
月	%m	以 01 - 12 形式表示月
月	%c	以 1 - 12 形式表示月
日	%d	以 01 - 31 形式表示日
日	%e	以 1 - 31 形式表示日
日	%D	以 th 后缀的形式表示日
日	%j	以 001 - 366 的形式表示日
周	%a	星期几对应的英文缩写
周	%W	星期几对应的英文全称
时	%H	以 00 - 23 的形式表示小时
时	%h	以 01 - 12 的形式表示小时
分	%i	以 00 - 59 的形式表示分钟
秒	%S	以 00 - 59 的形式表示秒
秒	%f	微秒
时分秒	%T	返回当前时刻的时分秒（ hh:mm:ss ）

- extract(unit from datetime)

unit	说明
year	年
month	月
day	日
hour	小时
minute	分钟
second	秒
week	周数，全年第几周

3. 日期和时间计算

- 向后偏移：date_add(date,interval num unit)
- 向前偏移：date_sub(date,interval num unit) 或 date_add(date,interval -num unit)
- 两个日期之差：datediff(end_date,start_date)
- 日期可以进行比较

八、数据分组与数据透视表

- 将表按照某列或某几列进行分组时，只需要在表后面通过 group by 指明具体的列名即可
- 除参加聚合运算的列外，要在 select 中查询的列必须先通过 group by 进行分组
- group by 后面的列名必须是原始表中的列名，不能是 select 过程中起的别名，也不能把聚合键之外的列名卸载 select 子句之中，having 的限制亦相同
- 使用 having，后面的聚合函数可以使用别名
- group_concat() 函数的作用是对组内的字符串进行连接，一般需要与 group by 结合使用
- 使用 group by 和 case when 组合的形式，可以实现数据透视表

九、窗口函数

- 只能写在 select 子句中

1. 序列函数

- ntile(n) 函数：主要用于对整张表的数据进行切片分组，默认是对表不进行任何操作之前进行切片分组
- row_number() 函数：用来排序，不会出现重复值，如果有两个相同的值，按照表中存储的顺序来生成行数
- lag(列名,n)：让数据向后移动
- lead(列名,n)：让数据向前移动
- first_value(列名)：截至当前行的第一个
- last_value(列名)：截至当前行的最后一个

2. 参数

- 截止到之前 n 行：rows n preceding
- 截止到之后 n 行：rows n following
- 当前记录的前后 n 行：rows between n preceding and n following

3. grouping 运算符（与 group by 结合使用）

- rollup：同时得到合计和小计，MySQL 8.x 中为 with rollup
- cube：多维聚合
- grouping sets：只取出合计和小计
- grouping：用来判断超级分组记录的 null，产生 null 时返回 1， 否则返回 0，常配合 case when 和 cast 使用

十、多表连接

1. 表的横向连接（合并）

- 用 on 来指明两张表中的公共列
- left join：左连接，以左表为主表，查找不到只保留左表信息，其余用 null 填充
- right join：右连接，以右表为主表，查找不到只保留右表信息，其余用 null 填充
- inner join：内连接，取交集
- outer join：外连接，去并集，MySQL 暂不支持，可以用左连接和右连接相结合（配合 union ）的方式来代替

2. 表连接的类型

- 一对一
- 一对多：程序会自动把一对多中没有重复值的列复制成多条记录
- 多对多：笛卡尔积，m * n，corss join
- 在实际工作中，要尽量避免一对多及多对多情况的出现

3. 多张表连接

- 公共列可以是多列，列与列之间直接在 on 后面用 and 连接

4. 标的纵向连接（追加）

- union：纵向连接后去重
- union all：纵向连接后不作任何处理，程序运行效率更高
- 不同表之间列的顺序要保持一致

5. 表的集合

- 表的交集：intersect
- 表的乘法：cross join
- 表的减法（左表减去两表的交集部分）：except

十一、子查询

*内层的查询为子查询，外层的查询为主查询

- 可分为 select 子查询、from 子查询、where 子查询
- 在细分组内进行比较时，需要使用关联子查询，结合条件一定要写在关联子查询中

十二、视图

- 视图即是临时表，多重视图嵌套会降低 SQL 的性能
- 子查询就是一次性视图，尽量避免使用多层嵌套子查询
- 定义视图时不能使用 order by 子句
- 通过汇总得到的视图无法进行更新

1. 创建视图

```
create view 视图名 ( 视图列1,视图列2,……,视图列n ) as select 语句
```

2. 删除视图

```
drop view 视图名 ( 视图列1,视图列2,……,视图列n ) ( cascade )
```

十三、SQL查询的执行顺序

1. 关键词

关键词	说明
select	指明要查询的列
from	指明要从哪张表查询
where	筛选表中满足条件的数据
group by	指明要按哪些列进行分组
having	筛选分组后满足条件的数据
order by	指明要按哪些列进行排序
limit	限制输出的行数

2. 执行顺序

- from —> where —> group by —> having —> select —> order by —> limit

十四、变量设置

- set @变量名 = 值
- select @变量名:= 值

- 如果使用 select 关键词进行变量赋值时，不可以直接使用 =

十五、DDL

- drop table 会把表完整删除，delete from table (where 条件) 会留下标的结构

1. 创建表

```
create tabel 表名 (
    列名1 数据类型 约束 comment 注释
    列名2 数据类型 约束 comment 注释
    ……
    列名n 数据类型 约束 comment 注释)
```

- MySQL 中常用的数据类型

数据类型	说明
int	整型，适用于一般位数的整数
bigint	极大整型，适用于超大位数的整数
float	浮点型
char	定长字符串，不能超过规定的最大字符长度
varchar	可变长字符串，会根据存入的字符串长度进行调节
date	日期类型
datetime	日期时间类型

- 主键约束：not null、 primary key(列1，列2，……，列n)、 default、 check 、自增、不可重复等

2. 向表中插入数据

```
insert into 表名 ( 列名1，列名2，……，列名n ) values
    (valueA1,valueA2,……,valueAn ),
    (valueB1,valueB2,……,valueBn ),
    ……
```

3. 修改表中的数据

- 新增列：alter table 表名 add 列名 数据类型
- 删除列：alter table 表名 drop 列名
- 修改数据类型：alter table 表名 modify 列名 新数据类型
- 修改表名：alter table 原表名 rename 新表名

4. 删除表

- drop table 表名

5. 复制表数据

```
create table 表名 按时 select 语句
```

6. 数据更新

```
update table 表名 set column = 表达式 where 条件
```

7. 创建事务

```
begin / start transaction
    DML 语句
    ……
commit / rollback
```