

数据准备



数据获取

知识点讲解

常用数据库

- **SQL server:** 在windows上运行，系统比较稳定，主要使用在银行和电信，国企类
- **MySQL:** 体积小、速度快、成本低，开源，中小企业会较多使用MySQL
- **Oracle:** 能所有主流平台上运行，性能高，价格比较贵，大公司使用较多的为Oracle
- **SQL语法:** mysql, sql server, oracle, postgresql等

数据获取方式

Python连接mysql

```
import pymysql
# 1.数据连接
conn = pymysql.connect(
    host='127.0.0.1',
    port=3306,
    user='yourname',
    password='yourpassword',
    db='test',
    charset='utf8')

# 2.创建游标
cursor = conn.cursor()

# 3.执行sql语句
cursor.execute("select * from ad_test
limit 10")
cursor.fetchall()
```

R连接mysql

```
library('DBI')
library(RMySQL)
con <- dbConnect(
    MySQL(),
    host="127.0.0.1",
    port=3306,
    username="yourname",
    password="yourpassword",
    dbname = "test"
)

dbSendQuery(con,'SET NAMES gbk')
dbGetQuery(con, "select * from ad_test
limit 10")
```

R连接SQL Server
library(RODBC)

安装: Navicat Premium



数据函数

聚合函数

聚合函数

示例代码

聚合函数AVG、SUM、COUNT、MIN/MAX 等

GROUP BY ROLLUP

GROUP BY CUBE

GROUP BY GROUPING SETS

(case when xx then xx else null end)

转换函数

转换函数

示例代码

---注意转化函数的除法问题

```
select 3/2
```

```
select cast(3 as float)/2
```

```
select 3*1.0/2
```

```
select cast(3*1.0/2 AS decimal(9,2))
```

--计算百分比

```
select cast(cast(3*100.0/2 AS decimal(9,1)) as  
varchar)+'%'
```

日期和时间函数

日期和时间函数

示例代码

```
SELECT current_date() as now,
```

```
# 日期的时间差
```

```
SELECT
```

```
    date_add('2020-12-31',interval -1 day) as yesterday,
```

```
    DATEDIFF('2020-12-31','2020-12-30') as diff,
```

```
    STR_TO_DATE('2020-01-01','%Y-%m-%d') as
```

```
strdate1,
```

```
    STR_TO_DATE('2020.01.01','%Y.%m.%d') as
```

```
strdate2
```

```
## 日期转化为时间
```

```
select unix_timestamp('2020-01-01 12:00:00')
```

```
## 时间戳转化为日期
```

```
select from_unixtime (0)
```

```
select from_unixtime (1577851200)
```


窗口函数

窗口函数

示例代码

常用窗口函数：
Row_number
Rank
Dense_rank
Partition by
其他窗口函数：
lag()和lead()取前N行或后N行，
first_values和last_values(取第一行和最后一行)

字符串函数

字符串函数

示例代码

```
select
  substring('广东省广州市天河区', 7), #返回字符串从start
位置到结尾的字符串##
  substring('素材创意01文案10', 1,6), #返回字符串A从
start位置开始, 长度为len的字符串
  left('广东省广州市天河区',2) AS province,
  right('广东省广州市天河区',3) AS area,
  insert('广东省广州市天河区',4,3,'珠海市') AS replace_city,
  replace('广东省广州市天河区','广州市','珠海市') AS
replace_city
```

字符串函数

模式	描述
^	匹配输入字符串的开始位置。如果设置了 RegExp 对象的 Multiline 属性，^ 也匹配 '\n' 或 '\r' 之后的位置。
\$	匹配输入字符串的结束位置。如果设置了 RegExp 对象的 Multiline 属性，\$ 也匹配 '\n' 或 '\r' 之前的位置。
.	匹配除 "\n" 之外的任何单个字符。要匹配包括 '\n' 在内的任何字符，请使用像 '[\n]' 的模式。
[...]	字符集合。匹配所包含的任意一个字符。例如， '[abc]' 可以匹配 "plain" 中的 'a'。
[^...]	负值字符集合。匹配未包含的任意字符。例如， '[^abc]' 可以匹配 "plain" 中的 'p'。
p1 p2 p3	匹配 p1 或 p2 或 p3。例如， 'z food' 能匹配 "z" 或 "food"。'(z f)ood' 则匹配 "zood" 或 "food"。
*	匹配前面的子表达式零次或多次。例如， zo* 能匹配 "z" 以及 "zoo"。* 等价于 {0,}。
+	匹配前面的子表达式一次或多次。例如， 'zo+' 能匹配 "zo" 以及 "zoo"，但不能匹配 "z"。+ 等价于 {1,}。
{n}	n 是一个非负整数。匹配确定的 n 次。例如， 'o{2}' 不能匹配 "Bob" 中的 'o'，但是能匹配 "food" 中的两个 o。
{n,m}	m 和 n 均为非负整数，其中n <= m。最少匹配 n 次且最多匹配 m 次。

加密函数

加密函数

示例代码

```
SELECT MD5('123456') #按md5模式加密,不可逆  
SELECT SHA('123456'); #按sha256模式加密,不可逆
```

```
SELECT AES_ENCRYPT('123456','8888'); #加密, 可逆  
SELECT  
AES_DECRYPT(AES_ENCRYPT('123456','8888'),'7777');  
#解密, 可逆, 密码错误  
SELECT  
AES_DECRYPT(AES_ENCRYPT('123456','8888'),'8888'); #  
解密, 可逆, 密码正确
```

```
SELECT ENCODE('123456','8888'); #加密, 可逆  
SELECT DECODE(ENCODE('123456','8888'),'7777'); #解  
密, 可逆, 密码错误  
SELECT DECODE(ENCODE('123456','8888'),'8888'); #解  
密, 可逆, 密码正确
```

其他函数

其他函数

示例代码

```
--coalesce函数:  
select COALESCE('abc', null,'dd',null), COALESCE(null,  
null,'dd',null)
```

```
---case 函数
```

```
--组合函数
```

```
union 和union all
```

```
--随机抽样
```

```
select rand()
```

```
--查询列名在某个表
```

```
information_schema.columns
```

```
select table_schema,table_name from
```

```
information_schema.columns where column_name =  
'city'
```

```
【WITH 语句】
```



指标计算-案例1

案例说明

得分表【scores_table】

id	name	subject	teacher_id	scores
1	赵雷	语文	1001	95
2	王菊	数学	1002	92
3	王风	物理	1003	90
4	赵云	数学	1002	98
5	赵雷	数学	1002	83
6	赵兰	语文	1001	78
7	郑竹	物理	1003	53
8	周虹	化学	1003	82
9	王菊	语文	1001	65
10	郑如	生物	1002	61

用户地域表【user_ip】

name	province	city	mobile
赵雷	广东	广州	18826307000
赵兰	上海市	上海	18821975313
王菊	广东省	深圳	18828928543
王电	北京	北京	18822900746
赵云	广西	阳朔	18822621535
周梅	广西省	桂林	18852653411
郑如	新疆	吐鲁番	1882426917

老师表【teacher_table】

teacher_id	teacher	subject
1001	赵亮	语文
1002	何达	数学
1003	赵心	物理
1003	赵心	化学
1002	何达	生物
1004	张健	数学

1.计算各个老师所教学生得分的最大值、最小值、每位老师所教的全部科目(此列为数组),
ps (要求没有学生成绩的老师得分显示为空) ;
列名 (teacher_id,teacher,max_scores,min_scores,subjects)

2.现发现得分表中“赵兰”的语文成绩实际上是“赵云”的，请对得分表进行改正，
并在得分表新增一个列 (row_scores) ，用来统计每个用户的学科排名情况
【比如赵雷语文成绩是1，数学成绩是2，王菊语文成绩是2，数学是1】；

3.计算名字姓“王”的学生在全部学生中的占比（保留两位小数，用%表示）以及90分以上的人数；

4.新增用户地域表的列：
province_1 （要求:把province列的"省"和"市"删除）
address （要求:合并province_1和city的字符串，且中间用'-'隔开,如果两列字符串相同，则无需合并。）
type （要求：手机尾号后三位，大于500，定义为1，小于500定义为2，空值定义为3）

5.统计手机尾号不等于000的所有用户（包括null），其每个用户的总成绩得分。(Name+scores)

案例说明

实现SQL

第一小题

```
select
    t1.teacher_id ,
    max(t2.scores) max_scores ,
    min(t2.scores) min_scores,
    group_concat(distinct t1.subject) as subjects
from
teacher_table t1
left join scores_table t2
on t1.teacher_id =t2.teacher_id
group by teacher_id
```


案例说明

实现SQL

第二小题

```
with t1 as (  
    select (case when name = '赵兰' then '赵云'  
    else name end )as name ,  
    subjects,  
    teacher_id,  
    scores  
    from scores_table  
)  
select * ,row_number()over(partitions by subject  
order by scores desc) as row_scores  
from t1
```

案例说明

实现SQL

第三小题

```
select
    count(distinct case when substr(name,1,1)='
王'then name else null end )/count(distinct name)
as "王姓占比",
    count(distinct case when scores >90 then
name else null end )/count(distinct name) as "90分
以上的人数"
from scores_table
```

案例说明

实现SQL

第四小题

```
select *,
        replace(replace(province,'市',''),'省','') as
province_1,
        case when replace(replace(province,'市',''),'省','')=city then city
              else concat(replace(replace(province,'市',''),'省',''),'-',city) end as address,
        case when mobile is null then 3 when
cast(substr(mobile,-3) as SIGNED)<500 then 2 else
1 end  as type
from user_ip
```

案例说明

实现SQL

第五小题

```
select
    a.name,
    coalesce(b.scores,0) as scores
from
    (select name from user_ip where
    substr(mobile,-3) !='000' or mobile is null) a
left join
    (select name ,sum(scores) as scores from
    scores_table group by name)b
    on a.name=b.name
```

谢谢观看