

第3单元 SQL 数据查询及统计分析

数据查询及统计分析是实际业务处理中的一项重要内容,诸如数据检索、排序、分类统计和数据合并等都属于数据查询及统计分析的业务范畴。

简单的数据检索问题可以充分利用 SQL 的基本 Select 语句来解决;常规的数据排序和分类统计一般由 Select 语句加排序(Order By)和分组(Group By)短语实施;比较复杂的任务需要借助 SQL 的语句嵌套、合并或视图进行综合设计。

本单元以人才招聘数据库为实例,实际演练 SQL 语句的设计方法和执行过程,使读者通过案例理解和掌握 SQL 语句的功能、设计方法和技巧,练就数据检索和统计分析的技能,提升用数据库技术解决较复杂问题的能力。

3.1 Select 语句的编辑和运行环境检测

有多种途径可以编辑和运行 SQL 语句,例如,在 MySQL 命令窗口中,可以输入、编辑和运行 SQL 语句,查看语句的运行效果,也可以通过 phpMyAdmin 可视化管理工具的 SQL 选项卡中的 SQL 语句编辑窗口生成、输入、编辑和运行 SQL 语句,查看和进一步编辑 SQL 语句运行的结果数据。

从操作方便和灵活性方面来看,phpMyAdmin 可视化管理工具更为实用一些。

一、实验目的

充分利用 phpMyAdmin 可视化数据库管理工具,学习生成、编辑和运行 SQL 语句的过程和方法,掌握设计和运行 MySQL 语句的一种操作环境和实际应用技能。

二、实验任务

- (1) 浏览岗位表 gwb 中除公司名称以外各列的全部数据记录。
- (2) 查看招聘超过 1 人的岗位编号、岗位名称、人数和岗位要求 4 列信息。

三、任务分析

两个实验任务的数据源都是来源于单个数据表,并且输出各列的数据都可以由数据表中的字段直接提取,因此,比较简捷的设计方法是通过 phpMyAdmin 的 SQL 语句编辑窗口生成 Select 语句,在此基础上再进行少量的修改和优化便可完成任务的设计要求。

四、预备知识

(1) 基本 Select 语句: Select <表达式表>From <数据源名>Where <条件>。

(2) SQL 语句的编辑与执行环境: 在 phpMyAdmin 的主页,单击选择当前数据库名(如 rczp)→数据表名(如 gwb)→代码编辑工具“编辑”,进入 SQL 语句的编辑窗口,再单击相关的按钮,可以生成简单的 SQL 语句。例如,单击 SELECT 按钮,生成的 Select 语句如图 3-1 所示。



图 3-1 SQL 语句编辑窗口

在此基础上,可以进一步编辑、完善、优化和执行 Select 语句。所谓优化就是按照 SQL 的语句规则,去掉语句中多余的信息,使语句变得更简洁和清晰,运行速度更快,节省存储空间,但仍然保留语句的功能。

五、技能点

(1) 生成 Select 语句: 恰当地运用工具,生成接近完成任务的 Select 语句。

(2) 编辑 Select 语句: 在系统生成的 Select 语句基础上, 删除、增加或修改相关内容, 使之成为完成任务的 Select 语句。

(3) 调试 Select 语句: 运行 Select 语句出错或结果不正确时, 能够发现和纠正错误。

(4) 执行 Select 语句: 检验 Select 语句的运行结果是否满足任务的要求。

六、注意事项

(1) 在系统生成的 Select 语句中, 数据库、数据表和字段名称均用左单引号“'”(“~”键的下档符号)括起来, 不能用右单引号“'”或双引号“”代替该符号。在上述名称中不含标点符号的情况下, 左单引号可以省略不写。例如, 图 3-1 中的 SQL 语句可以优化成语句:

```
SELECT 岗位编号, 岗位名称, 最低学历, 最低学位, 人数, 年龄上限, 年薪,  
笔试成绩比例, 笔试日期, 聘任要求, 公司名称 FROM gwb WHERE 1
```

(2) Select 语句中出现的数据库、数据表或字段名以外的符号, 一律以半角方式输入, 如 Select、From、各种引号、逗号、圆点、括号和分号等。英文字母不区分大小写。

(3) 一条语句可以分多行编写, 但不能在一个完整项(如语句名、短语名、数据库名、数据源名、字段名、函数调用或数据等)的中间分行。

七、实验步骤

1. 浏览岗位的相关信息

(1) 进入 SQL 语句编辑窗口: 在 phpMyAdmin 的主页, 单击数据库名“rczp”→数据表名“gwb”→代码编辑工具“编辑”。

(2) 生成 SQL 的 Select 语句: 单击 SQL 语句编辑窗口中的 SELECT 按钮, 生成的语句如图 3-1 所示。

(3) 编辑 Select 语句: 在 SQL 语句编辑框中, 删除左单引号、公司名称和 WHERE 1 等多余信息。修改后的语句如下:

```
SELECT 岗位编号, 岗位名称, 最低学历, 最低学位, 人数, 年龄上限, 年薪,  
笔试成绩比例, 笔试日期, 聘任要求 FROM gwb
```

(4) 执行 Select 语句: 单击 SQL 语句的编辑窗口中的“执行”按钮, 在 Select 语句的执行结果窗口(如图 3-2 所示)中可以观察到运行结果。

(5) 如果执行 Select 语句出错或者运行结果不符合要求, 则需要再单击 SQL 语句编辑窗口中的工具, 回到编辑窗口再次编辑、运行 Select 语句。

2. 查看招聘超过 1 人的岗位情况

(1) 对前述 Select 语句进行适当取舍和扩充, 最后设计的语句如下:

```
SELECT 岗位编号, 岗位名称, 人数, 聘任要求 FROM gwb WHERE 人数>1
```

(2) 单击 SQL 语句编辑窗口中的“执行”按钮, 执行效果如图 3-3 所示。

执行的SQL语句 设置显示行数 执行结果的数据行 SQL语句编辑窗口的工具



数据编辑工具组

图 3-2 Select 语句的执行结果窗口

岗位编号	岗位名称	人数	聘任要求
A0002	银行柜员	5	计算机二级, 笔试: 金融+会计学
A0003	律师	3	有驾照, 笔试经济法+金融
A0004	会计	3	笔试经济学+金融+计算机
B0001	经理助理	3	笔试: 经济学+人力资源
B0002	理财师	12	笔试: 经济法+财务管理
B0003	岗前培训师	2	笔试经济学+金融+计算机

图 3-3 招聘超过 1 人的岗位情况

八、思考题

(1) 在 MySQL 命令行和 phpMyAdmin 的 SQL 语句编辑窗口中都可以编辑和执行 Select 语句,各自的特点是什么?

(2) 进入 phpMyAdmin 的主页之前,要进行哪些操作? 执行数据查询的用户应该具有哪些权限? 当没有数据查询权限的用户执行 Select 语句时,系统将会如何反应?

3.2 查询语句的表达式设计

表达式是语句中完成各种计算和逻辑判断任务的重要工具,常数、变量(字段)和函数都是基本表达式,要完成较复杂的计算任务,往往需要利用运算符或谓词进一步连接表达式,设计更复杂的表达式。

一、实验目的

学习各类表达式的设计方法和手段,掌握表达式的用途和基本要素,特别是各类运算符、谓词及其相关函数的基本作用和功能,学会设计能解决实际问题的各类表达式。

二、实验任务

- (1) 检索年薪为 10~12 万元、聘任要求包含“经济”的岗位信息。
- (2) 检索岗位名称中含有“会计”或“行”字的岗位信息,输出信息包括岗位编号、岗位名称、年薪、人数和聘任要求 5 列信息,其中每个岗位的年薪后加“万元”两个字,列标题为“基础年薪”。
- (3) 检索未来 30 天内笔试的岗位信息,输出内容包括岗位编号、岗位名称、最低学历、笔试日期和聘任要求。其中最低学历(字符编码)转换成学历名称,对应的列标题为“学历要求”。

三、任务分析

3 个任务都需要设计数据检索条件,主要设计 Select 语句的 Where <条件表达式>。条件表达式由关系运算、逻辑运算、谓词运算及其相关函数组成,运算结果为逻辑真(非 0)或假(0)。第 1 个任务用谓词 Like 和 Between 设计条件表达式;第 2 个任务用谓词 RLike 和 Like 均可以设计条件表达式,但用 RLike 效果更佳;最后一个任务需要用 DateDiff 函数和关系运算设计条件表达式。

对于 Select <表达式>(计算列),设计第 2 个任务时,需要引用 Concat 函数;设计第 3 个任务时,需要借用 ELT 函数实现相关的任务要求。

四、预备知识

- (1) **谓词区间判断 Between**: X Between Y And Z,等效于 $X \geq Y$ And $X \leq Z$ 。
- (2) **谓词匹配运算 Like**: <字符串表达式 1>Like <字符串表达式 2>,字符串表达式 2 中可含匹配符号“%”和“_”,表示其位置任意多个或一个符号。
- (3) **谓词选择匹配运算 RLike**: <字符串表达式>RLike <子串 1>[|<子串 2>... |<子

串 n>],一般用于分析字符串表达式的值中是否含有子串之一。

(4) **DateDiff(D1,D2)函数**: 计算 D1 与 D2 两个日期之间的天数。如果第一个参数为系统日期函数 CurDate(),则可以计算 D2 到调用该函数当天的天数。

(5) **Concat(S₁[,S₂···,S_n])**: 字符串连接函数。返回 n 个字符串连接后的字符串,其中 S_i也可以是数值型数据。

(6) **ELT(N,S₁[,S₂,···,S_m])**: 数据编码转换成名称的函数。

五、技能点

(1) **条件表达式设计**: 恰当地运用各类运算符和函数设计 Where <条件表达式>。

(2) **计算列设计**: 正确设计 Select <表达式>,同时可以为计算列起列名称及标题。

六、注意事项

(1) 在调用函数时,参数要写在小括号内。即使没有参数,小括号也不能省略。

(2) 在 Like 和 RLike 运算符右侧的字符串表达式中,所包含的“%”“_”“|”不是要检索的符号,它们的含义是匹配符号或分隔符号。

(3) 调用 ELT 函数时,第一个参数是数值或数字串型表达式;其他参数为对应编码(从 1 开始)名称的字符串型表达式。

(4) 调用 DateDiff 函数时,两个参数均为日期型表达式,日期型常数要用单引号或双引号引起来。

(5) Where <条件表达式>中可以包含字段名、常数和普通函数,但不能使用 As 定义的列名和有关数据统计(聚类)函数(如 AVG、MAX 和 Sum 等)。

(6) 用<表达式>As <列名称>为计算列命名时,As 左右至少各有一个空格,省略 As 时,表达式与列名称之间至少有一个空格。

七、设计步骤

1. 检索年薪为 10~12 万元、聘任要求包含“经济”的岗位信息

(1) 在 phpMyAdmin 的主页,单击数据库名“rczp”→数据表名“gwb”→代码编辑工具“编辑”。

(2) 在编辑框中系统生成的语句最后再加短语: WHERE 年薪 BETWEEN 10 AND 12 AND 聘任要求 LIKE "%经济%",最后设计的语句如下:

```
SELECT * FROM gwb
WHERE 年薪 BETWEEN 10 AND 12 AND 聘任要求 LIKE "%经济%"
```

(3) 单击 SQL 语句的编辑窗口中的“执行”按钮,执行效果如图 3-4 所示。

2. 检索岗位名称中含有“会计”或“行”字的岗位信息

与上述设计和执行过程相似,满足任务要求的语句如下:

岗位编号	岗位名称	最低学历	最低学位	人数	年龄上限	年薪	笔试成绩比例	笔试日期	聘任要求	公司名称
A0001	行长助理	3	2	1	24	11	70	2022-10-14	有驾照, 笔试经济学+金融	工商前进支行
A0004	会计	3	3	3	35	10	60	2022-10-10	笔试经济学+金融+计算机	工商前进支行
B0001	经理助理	5	5	3	30	12	50	2023-10-21	笔试: 经济学+人力资源	腾讯总公司
B0003	岗前培训师	3	3	2	35	10	50	2023-05-08	笔试经济学+金融+计算机	工商前进支行

图 3-4 年薪为 10~12 万元、聘任要求包含“经济”的岗位信息

```
SELECT 岗位编号, 岗位名称, CONCAT(年薪, "万元") AS 基础年薪, 人数, 聘任要求 FROM gwb
WHERE 岗位名称 RLIKE "会计|行"
```

执行效果如图 3-5 所示。

岗位编号	岗位名称	基础年薪	人数	聘任要求
A0001	行长助理	11万元	1	有驾照, 笔试经济学+金融
A0002	银行柜员	10万元	5	计算机二级, 笔试: 金融+会计学
A0004	会计	10万元	3	笔试经济学+金融+计算机

图 3-5 岗位名称中含有“会计”或“行”字的岗位信息

3. 检索未来 30 天内笔试的岗位信息

- (1) 单击 SQL 语句编辑窗口中的“清除”按钮。
- (2) 在 SQL 语句编辑框中输入如下语句:

```
SELECT 岗位编号, 岗位名称,
ELT(最低学历, '无要求', '专科', '本科', '研究生', '博士') AS 学历要求,
笔试日期, 聘任要求 FROM gwb
WHERE DateDiff(笔试日期, CurDate()) BETWEEN 0 AND 30
```

- (3) 单击 SQL 语句编辑窗口中的“执行”按钮, 执行效果如图 3-6 所示。

岗位编号	岗位名称	学历要求	笔试日期	聘任要求
D0001	计算机工程师	本科	2022-09-18	计算机相关专业本科, 英语4级

图 3-6 未来 30 天内笔试的岗位信息

八、思考题

- (1) 谓词 Between 能否对其他数据类型(如字符串、日期等)的数据进行运算?
- (2) 谓词 RLIKE 和 LIKE 从功能方面有什么区别? 在上述 Select 语句中, 如果将 RLIKE 换成 LIKE, 或者将 LIKE 换成 RLIKE, 则应该如何设计相关的条件表达式, 同样确

保完成任务？

(3) 用 IF 和 ELT 函数都可以将数据编码转换成名称,在什么情况下用 ELT 函数更合适? 如果将性别码(1 表示男,2 表示女)转换成汉字,如何调用这两个函数?

3.3 多个数据源的查询设计

数据源是指运行查询时系统获取数据的渠道和对象。在 MySQL 数据库管理系统中,数据源可为数据表或数据视图(虚拟表)。

一、实验目的

学习和巩固多个数据源(数据表或视图)的连接方法,掌握数据源之间的常用连接类型、功能以及必要条件,学会从多个数据源中同时提取所需要的数据,深入理解数据源之间关联的作用和必要性。

二、实验任务

- (1) 输出每个岗位的岗位编号、岗位名称、公司名称和公司地址。
- (2) 输出本年度笔试、有人申报的岗位编码和岗位名称,多人申报的岗位仅输出一次。
- (3) 输出每个岗位的岗位编号、岗位名称,申报人员的身份证号、姓名和总分,申报同岗位人员连续输出,并按总分由高到低排序。
- (4) 输出目前还没有人申报的岗位,内容包括岗位编号和岗位名称。

三、任务分析

第 1 个任务要求输出的岗位编号、岗位名称、公司名称都来源于岗位表(gwb),但公司地址来源于公司表,因此,需要同时连接这两个表。

第 2 个任务要求输出岗位编码和岗位名称,输出条件之一需要笔试日期,可以从岗位表(gwb)中得到这些字段。但是,输出条件之二是有人申报的岗位,也就是说仅输出岗位成绩表(gwcjb)中存在的岗位。因此,还需要 gwb 和 gwcjb 两个表进行连接才能完成任务。另外,还要加 Distinct 短语,避免输出重复的数据行。

第 3 个任务需要 gwb 中的岗位编号、岗位名称和笔试成绩比例,从 ypryb 中得到身份证号和姓名,从 gwcjb 中提取笔试成绩和面试成绩,因此,完成这个任务需要 gwb、ypryb 和 gwcjb 三个表连接。此外,还需要 Order By 短语对岗位编号和总分两个关键字进行排序。

第 4 个任务依据是否有人申报来决定输出岗位信息,实质要输出岗位表(gwb)与 gwcjb 无关联的记录,连接类型可以选择左连接 Left Join 或右连接 Right Join,并与 IsNull 函数结合操作。

四、预备知识

从多个数据源进行数据查询,需要数据源之间进行连接。在 MySQL 数据库管理系统中,数据源之间的连接方式有如下两种。

(1) **From <数据源名表>Where <连接条件>**: 数据源名表是用半角逗号“,”分隔的多个数据源名,数据源名可为数据表名或视图名;连接条件通常是两个数据源关联关键字的等值表达式。

(2) **From <数据源名 1>[<连接类型><数据源名 2>][On <连接条件>]**: 连接类型常用 Join(内连接)、Left Join(左连接)、Right Join(右连接)和 Natural Join(自然连接)4 种。连接条件通常也是关联关键字的等值表达式。此种连接方式通常要比第一种连接方式速度快。而 Natural Join 不用 On <连接条件>,实质是两个数据源中同名字段的等值连接。

五、技能点

(1) **连接类型选择**: 选择合适的连接方式和类型可以简化 Select 语句,提高数据查询速度。

(2) **连接条件设计**: 为数据源之间的连接设置表达式,表达式的值是决定数据源中记录连接成功与否的条件。

六、注意事项

(1) 两个或更多数据源中出现的字段名,其前必须加“数据源名.”,但连接类型为 Natural Join 时,不受此限制。

(2) On <连接条件>必须与某个连接类型(Join、Left Join 或 Right Join 等)结合使用,不能单独存在。

(3) Select 语句中各个数据源之间都要直接或间接地进行关联,否则,将产生不可预测的查询结果。

(4) 用 From <数据源名表>短语说明多个数据源,数据源之间的连接条件必须写在 Where <连接条件>中,不能写在 On <连接条件>中。同样,用 From <数据源名 1><连接类型><数据源名 2>短语,数据源之间的连接条件必须写在 On <连接条件>中,而不能写在 Where <连接条件>中。总之,两种连接方式不能组合使用。

七、设计步骤

(1) 输出每个岗位的岗位编号、岗位名称、公司名称和公司地址,可以用下列 3 种语句形式之一实现。

① 用 From <数据源名表>Where <连接条件>设计:

```
SELECT 岗位编号, 岗位名称, gwb.公司名称, 地址 FROM gwb, 公司表
WHERE gwb.公司名称=公司表.公司名称;
```

② 用 From <数据源名 1>Inner Join <数据源名 2>On <连接条件>设计:

```
SELECT 岗位编号, 岗位名称, gwb.公司名称, 地址 FROM gwb INNER JOIN 公司表
ON gwb.公司名称=公司表.公司名称;
```

③ 用 From <数据源名 1>Natural Join <数据源名 2>设计:

```
SELECT 岗位编号, 岗位名称, 公司名称, 地址 FROM gwb Natural JOIN 公司表;
```

隐含说明连接条件为“gwb.公司名称=公司表.公司名称”。

3 条语句输出的结果相同, 执行效果如图 3-7 所示。

(2) 输出本年度笔试、有人申报的岗位编号和岗位名称:

```
SELECT DISTINCT gwb.岗位编号, 岗位名称
FROM gwb INNER JOIN gwcjb ON gwb.岗位编号=gwcjb.岗位编号
WHERE year(笔试日期)=year(CurDate());
```

语句的执行效果如图 3-8 所示。

岗位编号	岗位名称	公司名称	地址
A0001	行长助理	工商前进支行	长春市高新区
A0002	银行柜员	工商前进支行	长春市高新区
A0003	律师	工商前进支行	长春市高新区
A0004	会计	工商前进支行	长春市高新区
B0001	经理助理	腾讯总公司	北京市中关村
B0002	理财师	腾讯总公司	北京市中关村
B0003	岗前培训师	工商前进支行	长春市高新区
D0001	计算机工程师	工商前进支行	长春市高新区

图 3-7 含公司地址的岗位信息

岗位编号	岗位名称
A0001	行长助理
A0002	银行柜员

图 3-8 本年度笔试的岗位信息

(3) 输出每个岗位的岗位编号、岗位名称、身份证号、姓名和总分:

```
Select 岗位编号, 岗位名称, 身份证号, 姓名,
笔试成绩 * 笔试成绩比例 / 100 + 面试成绩 * (1 - 笔试成绩比例 / 100) As 总分
From GWB Natural Join GWCJB Natural Join YPRYB
Order By 1, 总分 DESC ;
```

语句的执行效果如图 3-9 所示。

(4) 输出目前无人申报的岗位信息的语句如下:

```
SELECT gwb.岗位编号, 岗位名称
FROM gwb LEFT JOIN gwcjb ON gwb.岗位编号=gwcjb.岗位编号
WHERE ISNULL(身份证号);
```

语句的执行效果如图 3-10 所示。

岗位编号	岗位名称	身份证号	姓名	总分
A0001	行长助理	229901199305011575	赵明	85.5000
A0001	行长助理	11980119921001132X	王丽敏	79.5000
A0002	银行柜员	229901199503121538	刘德厚	81.5000
A0002	银行柜员	229901199305011575	赵明	75.5000
A0002	银行柜员	219901199001011351	郝帅	70.0000
B0001	经理助理	11980119921001132X	王丽敏	82.5000
B0001	经理助理	219901199001011351	郝帅	67.5000
B0001	经理助理	229901199305011575	赵明	55.0000

图 3-9 应聘人员的成绩信息

岗位编号	岗位名称
A0003	律师
A0004	会计
B0003	岗前培训师
D0001	计算机工程师

图 3-10 无人申报的岗位信息

八、思考题

- (1) 任务 2 的设计语句中,数据源之间的连接类型及条件是否也可以用自然连接来实现?如何修改 Select 语句才能用自然连接?自然连接有哪些优点?要使用自然连接,对定义表结构有哪些要求?
- (2) 如何修改任务 3 的 Select 语句,使之用内连接完成本任务?
- (3) 如何修改任务 4 的 Select 语句,使之用右连接完成本任务?
- (4) 在本实验的 Select 语句中,有几处出现“gwb.岗位编号”,哪条语句中可以省略“gwb.”?为什么其他位置不能省略“gwb.”?

3.4 数据统计分析设计

数据统计分析通常指数据排序、数据分组及其数据个数、最大值、最小值、合计和平均值等方面的统计分析。

一、实验目的

学习数据的排序和统计分析技术,掌握 SQL 语句在数据统计分析方面的作用、设计方法及要领,增强用计算机技术进行数据统计分析的意识,提升使用数据库技术解决实际应用问题的能力。

二、实验任务

- (1) 输出各个岗位的申报情况,包括公司名称、岗位编号、岗位名称、身份证号(申报人员)、姓名和笔试成绩。同一公司的,按岗位编号排序;同一岗位的,按笔试成绩由高到低排序。
- (2) 输出有人申报的每个岗位情况,包含岗位编号、岗位名称、申报人数,总分的最高

分、最低分及平均分,小数点后均保留一位。数据按申报人数由多到少进行排列。

(3) 输出缺额的岗位情况,包括目前还没有申报满额或没人申报的岗位编号、岗位名称、招聘人数、申报人数和缺额人数,按缺额的多少排列输出结果。

(4) 输出笔试平均成绩前三名的岗位编号、岗位名称、申报人数和平均成绩,平均成绩保留小数点后一位。

(5) 用 SQL-Select 语句输出各时期的 GDP 情况。内容包括时期、最高 GDP 亿元、平均 GDP 亿元、最高排名、平均排名、人均最高 GDP、人均平均 GDP、人均最高排名和人均平均排名。

三、任务分析

第 1 个任务是多个排序关键字的排序问题。第一个排序关键字为公司名称,第二个排序关键字为岗位编号,前两个排序关键字升序或降序均可。第三个排序关键字为笔试成绩,必须为降序排序方式。

第 2 个任务是分组统计及排序分析问题,分组关键字是岗位编号,排序关键字是申报人数。由于仅输出有人申报的岗位信息,因此设计需要 gwb 和 gwcjb 按岗位编号进行内连接或自然连接。另外,总分=笔试成绩 * 笔试成绩比例/100+面试成绩 * (1-笔试成绩比例/100)。

第 3 个任务由于要输出有人申报和无人申报的岗位情况,因此设计需要 gwb 和 gwcjb 按岗位编号进行左连接。此外,按要求仅输出没满额的岗位情况,因此,要对查询统计结果进一步筛选,需要用 Having <筛选条件>短语对查询结果再次筛选。

第 4 个任务是数据分组、排序和限制数据行的综合问题,需要用 Group By、Order By 和 Limit 短语综合设计。

第 5 个任务中,由于排名的值越小,排名越高,因此,语句中用 Min 函数计算最高排名。

四、预备知识

(1) **Order By <排序关键字>**: 设置多个关键字排序,只有前面的关键字值相同时才按后面的关键字排序,并且每个排序关键字后可以加 ASC(可省略)或 DESC 以控制结果数据行的升序或降序排列。

(2) **Group By <分组关键字>**: 将分组关键字值相同的数据记录统计成查询结果中的一行数据,每个分组关键字后也可以加 ASC(升序)或 DESC(降序)以控制结果数据行的排列顺序。

(3) **统计(聚类)函数**: Group By 短语经常与 AVG(平均值)、Count(记录个数)、Max(最大值)、Min(最小值)和 Sum(合计)等聚类函数结合,对分组关键字的值进行统计分析。

五、技能点

(1) **选择排序关键字**：Order By 短语用于设置排序关键字,当要求输出前 n 行数据时,要与 Limit<行数>短语结合。

(2) **选择分组关键字**：对于“每个……、各个……”一类的任务要求,一般要用 Group By 短语设置分组关键字。如果要进一步筛选掉统计结果中的部分数据行,还要加 Having <条件>短语。

(3) **调用统计函数**：在 Select 语句中可以调用统计函数,如果包含 Group By 短语,则基于每个分组值的记录进行统计,否则,对全部记录进行统计。

(4) **精准应用 Where <条件>与 Having <条件>**：在 Where <条件>中,通常写从数据源中提取数据或数据源之间的连接条件,不能包含聚类函数(如 Max、Min、Avg 等)和计算列(非数据源中的列),这类内容只能写在 Having <条件>中。

六、注意事项

(1) 当 Limit 与 Order By 短语结合时,输出结果的数据行数严格受到 Limit 短语的限制,因此,可能遗漏与最后一行同排序关键字值的其他数据行。

(2) 在含有 Group By 短语的 Select 语句查询结果中,只有统计列、分组关键字能唯一确定的列对分组关键字的值才有依存关系。

(3) 在 Where <条件>短语的表达式中,只能出现数据源中的列名及相关的运算符和函数,或者数据源之间的连接条件,但不能出现聚类函数和计算列。

七、设计步骤

(1) 输出各个岗位申报情况,设计语句如下:

```
SELECT 公司名称,岗位编号,岗位名称,身份证号,姓名,笔试成绩
FROM gwb NATURAL JOIN gwcjb NATURAL JOIN ypryb
ORDER BY 公司名称,岗位编号,笔试成绩 DESC;
```

运行该语句的输出结果如表 3-1 所示。

表 3-1 3 个排序关键字的查询结果

公司名称	岗位编号	岗位名称	身份证号	姓名	笔试成绩
工商前进支行	A0001	行长助理	229901199305011575	赵明	90
工商前进支行	A0001	行长助理	11980119921001132X	王丽敏	75
工商前进支行	A0002	银行柜员	229901199503121538	刘德厚	80
工商前进支行	A0002	银行柜员	229901199305011575	赵明	80
工商前进支行	A0002	银行柜员	219901199001011351	郝帅	70

表 3-3 缺额的岗位情况

岗位编号	岗位名称	招聘人数	申报人数	缺额人数
B0002	理财师	12	4	8
A0004	会计	3	0	3
A0003	律师	3	0	3
A0002	银行柜员	5	3	2
B0003	岗前培训师	2	0	2
D0001	计算机工程师	1	0	1

申报人数为 0, 表示无人申报, 是左连接 (LEFT JOIN) 的效果。

缺额人数由多到少排序

(4) 输出笔试平均成绩前三名岗位情况的语句如下:

```
Select 岗位编号, 岗位名称, Count(*) As 申报人数,
       ROUND(AVG(笔试成绩), 1) As 笔试平均分
From   GWB NATURAL JOIN GWCJB
Group By 岗位编号 ORDER BY 平均分 DESC LIMIT 3;
```

运行语句的输出结果如表 3-4 所示。

表 3-4 笔试成绩前三名的岗位情况

岗位编号	岗位名称	申报人数	笔试平均分
B0002	理财师	4	84.7
A0001	行长助理	2	82.5
A0002	银行柜员	3	76.7

(5) 输出各时期的 GDP 情况。

```
SELECT 事件名称 AS 时期,
       Max( GDP 亿元 ) AS 最高 GDP 亿元,
       Floor( Avg( GDP 亿元 ) ) AS 平均 GDP 亿元,
       Min( 排名 ) AS 最高排名,
       Avg( 排名 ) AS 平均排名,
       Max( 人均 GDP ) AS 人均最高 GDP,
       Floor( Avg( 人均 GDP ) ) AS 人均平均 GDP,
       Min( 人均排名 ) AS 人均最高排名,
       Floor( Avg( 人均排名 ) ) AS 人均平均排名
FROM   国民经济状况
GROUP BY 事件名称
ORDER BY 2 DESC;
```

语句中 Floor 是对相关参数取整数的函数, 运行该语句的输出结果如表 3-5 所示。

表 3-5 各时期的 GDP 情况

时期	最高 GDP 亿元	平均 GDP 亿元	最高排名	平均排名	人均最高 GDP	人均平均 GDP	人均最 高排名	人均平 均排名
十九大	1 210 207	1 054 648	2	2	85 698	74 822	63	75
十八大	832 036	700 763	2	2	59 592	50 741	90	95
十八大前	538 580	308 251	2	3.5	39 771	23 188	110	124

从统计结果表 3-5 可以看出,自中国共产党的十八大后(2013—2022 年)的 10 年期间,我国 GDP 一直处于世界第二位,人均 GDP 排名已经进入世界前百位,并且逐年稳步攀升。十九大后的 5 年(2018—2022 年)中,人均 GDP 排名有着更大幅度的提升,最高排名为 63 位,平均排名为 75 位。特别是二十大报告将数字经济作为我国构建现代化经济体系的重要引擎,预计在未来的 5 年中,人均 GDP 排名将稳步站在世界的平均线上(2022 年世界人均 GDP 为 1.2 万美元,排名 70 位),预期排名将逐年提升。

八、思考题

- (1) Order By 和 Group By 短语中均可以加 ASC 或 DESC 以控制结果数据行的顺序。在含有 Group By 短语的 Select 语句中,要使统计结果数据行排序,什么情况下可以不用 Order By 短语? 什么情况下必须用 Order By 短语?
- (2) 在完成第 3 个任务时,如何用右连接实现?
- (3) Where <条件>和 Having <条件>都能起筛选数据行的作用,二者有何区别? 各自适合什么情况?

3.5 SQL 语句的嵌套设计

一、实验目的

学习 SQL 语句的嵌套设计技术,掌握 SQL 语句的嵌套位置、运行过程和作用,以便利用嵌套技术解决更复杂的实际应用问题。

二、实验任务

- (1) 输出无人申报的岗位情况,包含按岗位编号排序输出没人申报的岗位编号、岗位名称和招聘人数。
- (2) 输出高于岗位笔试平均分的申报人员情况,包含身份证号、姓名、岗位名称和笔试成绩,申报相同岗位的,按笔试成绩由高到低排序。
- (3) 输出每个岗位的笔试成绩最高分和最高分获得者的情况,包含岗位编号、岗位名称、笔试成绩最高分、身份证号和姓名,按岗位编号升序输出。

- (4) 按姓名排序输出申报人员同名情况,包含身份证号、姓名、出生日期和性别。
- (5) 自动填写资格审核字段的值。资格审核是否通过的条件是:满足学历和学位要求,并且笔试和面试成绩均 60 分及以上。用一条 SQL 语句填写“资格审核”字段的值。
- (6) 删除没有申报任何岗位的应聘人员记录。

三、任务分析

第 1 个任务实质是查找 gwb 中有、而 gwcjb 中不存在的岗位,主查询用 gwb 输出岗位编号、岗位名称和招聘人数,在 Where 条件中用谓词 Not In 嵌套从 gwcjb 中查询岗位编号的子查询。也可以通过 gwb 与 gwcjb 的左连接查询实现。

第 2 个任务主查询需要 ypryb、gwb 和 gwcjb 3 个表连接输出身份证号、姓名、岗位名称和笔试成绩,在 Where 条件中设计对应岗位的笔试平均分子查询。

第 3 个任务的主查询需要 gwb、gwcjb 和 ypryb 3 个表连接输出岗位编号、岗位名称、笔试成绩(最高分)、身份证号和姓名,在 Where 条件中设计“笔试成绩=”对应岗位的笔试成绩最高分的子查询,或者设计“笔试成绩>=All”对应岗位的全部笔试成绩的子查询。

第 4 个任务的主、子查询的数据源都是 ypryb,主查询输出身份证号、姓名、性别和出生日期,第二代身份证号码的第 7 位至 14 位共 8 位表示出生日期,第 17 位为奇数表示男性,为偶数表示女性,因此,从身份证号码中可以获取应聘人员的出生日期和性别信息。在 Where 条件中用谓词 Exists 对子查询进行操作,子查询只查找与主查询当前记录的同名者人数或非本人的同名者。

第 5 个任务的主语句为更新 gwcjb 中资格审核(逻辑型)字段的 Update 语句,子语句为提取资格审核条件的 Select。

第 6 个任务的主语句从 ypryb 中删除记录,Where 条件中用谓词 Not In,子查询中取 gwcjb 中的身份证号即可。

四、预备知识

- (1) **子查询常用谓词**: [Not] In、All、Any|Some 和 [Not] Exists。
- (2) **子查询的嵌套位置**: 通常嵌套在以 Update、Delete 和 Select 为主语句的表达式中,具体可以嵌套在 Select <表达式>、Where <条件表达式>或 Update ... Set <字段名>=<表达式>中。

五、技能点

- (1) **子查询设计**: 与独立执行的 Select 语句有些差异,绝大多数子查询的结果为一个或一列数据;多数子查询中引用主 SQL 语句中的数据,因而不能独立执行。
- (2) **选择子查询运算符**: 子查询结果为一个数据(含 Null)时,可以视为一个普通数据,进行算术运算、比较运算、逻辑运算或作为函数的参数;当子查询结果为数据集时,必须用集合谓词运算符 [Not] In、All、Any|Some 或 [Not] Exists。

六、注意事项

- (1) 子查询语句必须用小括号括起来,嵌套在主 SQL 语句的表达式中。
- (2) 当子查询结果为数据集时,不能用常规运算符对其进行运算,否则系统出错。
- (3) 当主语句为 Update 或 Delete 时,子查询的数据源不能是主语句中正处理的数据表。

七、设计步骤

- (1) 输出无人申报的岗位情况。

- ① 用嵌套语句设计如下:

```
SELECT 岗位编号,岗位名称,人数 AS 招聘人数
      FROM gwb WHERE 岗位编号 NOT IN (SELECT 岗位编号 FROM gwcjb)
      ORDER BY 岗位编号;
```

- ② 用数据源左连接的语句设计如下:

```
SELECT gwb.岗位编号,岗位名称,人数 AS 招聘人数
      FROM gwb LEFT JOIN gwcjb ON gwb.岗位编号=gwcjb.岗位编号
      WHERE 身份证号 IS NULL
      ORDER BY 岗位编号;
```

运行上述语句输出的结果如图 3-11 所示。

- (2) 输出高于岗位笔试平均分的申报人员情况的语句如下:

```
SELECT 身份证号,姓名,岗位名称,笔试成绩
      FROM ypryb NATURAL JOIN gwcjb AS CJB NATURAL JOIN gwb
      WHERE 笔试成绩>(SELECT AVG(笔试成绩) FROM gwcjb WHERE CJB.岗位编号=gwcjb.岗位编号)
      ORDER BY 岗位编号,笔试成绩 DESC;
```

运行语句的输出结果如图 3-12 所示。

岗位编号	岗位名称	招聘人数
A0003	律师	3
A0004	会计	3
B0003	岗前培训师	2
D0001	计算机工程师	1

图 3-11 无人申报的岗位信息

身份证号	姓名	岗位名称	笔试成绩
229901199305011575	赵明	行长助理	90
229901199503121538	刘德厚	银行柜员	80
229901199305011575	赵明	银行柜员	80
11980119921001132X	王丽敏	经理助理	85
219901199001011351	郝帅	经理助理	75
11980119921001132X	王丽敏	理财师	90
219901199001011351	郝帅	理财师	89

图 3-12 高于岗位笔试平均分的申报人员情况

- (3) 输出每个岗位的笔试成绩最高分和最高分获得者情况。

- ① 用关系运算符“=”与子查询进行运算,语句设计如下:

```
SELECT 岗位编号, 岗位名称, 笔试成绩 AS 最高分, 身份证号, 姓名
      FROM gwb NATURAL JOIN gwcjb AS CJB NATURAL JOIN ypryb
      WHERE 笔试成绩=(SELECT MAX(笔试成绩) FROM gwcjb WHERE CJB.岗位编号=gwcjb.岗位
      编号)
      ORDER BY 岗位编号;
```

② 用谓词“>=All”与子查询进行运算, 语句设计如下:

```
SELECT 岗位编号, 岗位名称, 笔试成绩 AS 最高分, 身份证号, 姓名
      FROM gwb NATURAL JOIN gwcjb AS CJB NATURAL JOIN ypryb
      WHERE 笔试成绩>=ALL (SELECT 笔试成绩 FROM gwcjb WHERE CJB.岗位编号=gwcjb.岗位编
      号)
      ORDER BY 岗位编号;
```

运行上述语句输出的结果如图 3-13 所示。

岗位编号	岗位名称	最高分	身份证号	姓名
A0001	行长助理	90	229901199305011575	赵明
A0002	银行柜员	80	229901199503121538	刘德厚
A0002	银行柜员	80	229901199305011575	赵明
B0001	经理助理	85	11980119921001132X	王丽敏

图 3-13 每个岗位的笔试成绩最高分及最高分获得者

(4) 输出申报人员同名情况。

① 用非本人的同名者子查询设计如下:

```
SELECT 身份证号, 姓名, MID(身份证号, 7, 8) AS 出生日期,
      IF(MOD(MID(身份证号, 17, 1), 2) = "1", "男", "女") AS 性别
      FROM ypryb
      WHERE EXISTS (SELECT * FROM ypryb AS ryb
                    WHERE ypryb.姓名=ryb.姓名 AND ypryb.身份证号<>ryb.身份证号);
      ORDER BY 姓名;
```

② 用统计同名者人数的子查询设计如下:

```
SELECT 身份证号, 姓名, MID(身份证号, 7, 8) AS 出生日期,
      IF(MOD(MID(身份证号, 17, 1), 2) = "1", "男", "女") AS 性别
      FROM ypryb
      WHERE (SELECT COUNT(*) FROM ypryb AS ryb WHERE ypryb.姓名=ryb.姓名)>1
      ORDER BY 姓名;
```

运行上述语句输出的结果如图 3-14 所示。

身份证号	姓名	出生日期	性别
220104200210011548	李丽丽	20021001	女
229901199305011524	李丽丽	19930501	女
11980119921001132X	赵明	19921001	女
229901199305011575	赵明	19930501	男

图 3-14 申报人员同名情况

(5) 自动填写资格审核字段值的语句如下:

```
UPDATE gwcjb SET 资格审核 =  
    ( SELECT 最后学历>=最低学历 AND 最后学位>=最低学位  
      FROM gwb, ypryb WHERE  
        ypryb.身份证号=gwcjb.身份证号 AND gwb.岗位编号=gwcjb.岗位编号 )  
    AND 笔试成绩>=60 AND 面试成绩 >=60;
```

在主语句为 Update gwcjb 的子查询语句中,可以引用主语句中的数据(如笔试成绩和面试成绩),但子语句的数据源不能包含主语句中要更新的数据源(如 gwcjb)。

(6) 删除没有申报任何岗位的应聘人员记录的语句如下:

```
DELETE FROM ypryb  
    WHERE 身份证号 NOT IN (SELECT DISTINCT 身份证号 FROM gwcjb);
```

主语句删除 ypryb 中的部分记录,因此,同样子查询的数据源不能包含 ypryb。

八、思考题

(1) 在第 1 个任务的两种设计方法中,哪种方法运行速度更快?

(2) 将下列语句的执行结果与图 3-13 进行对比分析,此语句能否完成第 3 个任务,其原因何在?

```
SELECT 岗位编号,岗位名称,MAX(笔试成绩) AS 最高分,身份证号,姓名  
    FROM gwb NATURAL JOIN gwcjb NATURAL JOIN ypryb  
    GROUP BY 岗位编号;
```

(3) 在完成第 4 个任务的语句中,除了用 IF 函数输出性别外,还能用哪些函数实现这种功能? 修改语句中的相关内容,使之功能不变。

3.6 SQL 语句的合并(联合)设计

一、实验目的

测试多条 SQL 语句的合并效果,了解其功能范畴,掌握 SQL 语句的综合设计及应用技术,以便灵活地运用数据库技术进行数据维护和统计分析。

二、实验任务

(1) 备份 gwcjb。将岗位成绩表(gwcjb)中的数据备份到当前数据库的 cjb 表中,不需要主键。

(2) 将资格审核合格的申报信息保存到新表 shhg 中。审核合格表(shhg)中包含岗位编号、岗位名称、身份证号、姓名和总分 tinyint(3) 5 个字段。其中前 4 个字段与 gwb 或 ypryb 中对应字段的数据类型和宽度一致,岗位编号和身份证号共同组成 shhg 的主键。