《数据库技术实验》报告

|  |  |  |
| --- | --- | --- |
| **第 三 次实验** | **日期：2022-10-04** | **得分：** |
| **学号：** | **姓名：*Steven*** | **专业：智能科学与技术** |

一、实验目的

1. 掌握查询数据的SQL语句——Select语句

2. 掌握单表查询和连接查询

3. 了解聚集函数和GROUP BY…HAVING子句

二、实验内容

（一）实验题

1. 在Student\_Course数据库上验证课本上单表查询和连接查询的例子

2. 在World数据库的表Country\_Simple、City\_Simple、CL\_Simple上自行设计5个以上单表查询语句

3. 在World数据库的表Country\_Simple、City\_Simple、CL\_Simple上自行设计5个以上连接查询语句

（二）思考题

1. WHERE子句和HAVING子句的区别是什么？

三、实验过程及实验结果

1.

(1) 查询考试成绩<70的学生的学号（因为sc中没有不及格的）

|  |
| --- |
| SELECT DISTINCT Sno FROM SC WHERE Grade<70; |
| mysql> SELECT DISTINCT Sno FROM SC WHERE Grade<70;  +-------------+  | Sno |  +-------------+  | 20191060001 |  +-------------+  1 row in set (0.00 sec) |

(2) 查询全体学生情况，查询结果按所在系的系号升序排列，同一系中的学生按年龄降序排列。

|  |
| --- |
| SELECT \* FROM student ORDER BY Sdept ASC,Sage DESC; |
| mysql> SELECT \* FROM student ORDER BY Sdept ASC,Sage DESC;  +-------------+--------+------+------+--------------+  | sno | sname | ssex | sage | sdept |  +-------------+--------+------+------+--------------+  | 20191060001 | 张三 | 男 | 20 | 计算机 |  | 20191060003 | 王五 | 男 | 20 | 计算机 |  | 20191060005 | 钱七 | 女 | 18 | 计算机 |  | 20191060002 | 李四 | 女 | 19 | 通信工程 |  | 20191060004 | 赵六 | 男 | 19 | 通信工程 |  +-------------+--------+------+------+--------------+  5 rows in set (0.00 sec) |

(3) 查询选修1号课程的学生平均成绩

|  |
| --- |
| SELECT MAX(Grade) FROM SC WHERE Cno='1'; |
| mysql> SELECT MAX(Grade) FROM SC WHERE Cno='1';  +------------+  | MAX(Grade) |  +------------+  | 90 |  +------------+  1 row in set (0.00 sec) |

(4) 查询选修了三门以上课程的学生学号

|  |
| --- |
| SELECT Sno FROM sc GROUP BY Sno HAVING COUNT(\*)>3; |
| mysql> SELECT Sno FROM sc GROUP BY Sno HAVING COUNT(\*)>3;  +-------------+  | Sno |  +-------------+  | 20191060003 |  +-------------+  1 row in set (0.00 sec) |

(5) 查询选修2号课程且成绩在90分以上的所有学生的学号和姓名。

|  |
| --- |
| SELECT      student.Sname AS '姓名',      sc.sno AS '学号'  FROM student, sc  WHERE      student.Sno = sc.Sno AND sc.Cno = '2' AND sc.Grade > 90; |
| mysql> SELECT  -> student.Sname AS '姓名',  -> sc.sno AS '学号'  -> FROM student, sc  -> WHERE  -> student.Sno = sc.Sno AND sc.Cno = '2' AND sc.Grade > 90;  +--------+-------------+  | 姓名 | 学号 |  +--------+-------------+  | 王五 | 20191060003 |  +--------+-------------+  1 row in set (0.00 sec) |

(6) 以Student为主体列出每个学生的基本情况和选课情况。此时应注意的是，若没有选课信息，也应当将该学生的信息保留，这种情况应当使用外连接。

|  |
| --- |
| SELECT      student.sno, sname, ssex, sdept, cno, grade  FROM student      LEFT OUTER JOIN sc ON (student.sno = sc.sno); |
| mysql> SELECT  -> student.sno, sname, ssex, sdept, cno, grade  -> FROM student  -> LEFT OUTER JOIN sc ON (student.sno = sc.sno);  +-------------+--------+------+--------------+------+-------+  | sno | sname | ssex | sdept | cno | grade |  +-------------+--------+------+--------------+------+-------+  | 20191060001 | 张三 | 男 | 计算机 | 1 | 68 |  | 20191060001 | 张三 | 男 | 计算机 | 2 | 76 |  | 20191060001 | 张三 | 男 | 计算机 | 3 | 80 |  | 20191060002 | 李四 | 女 | 通信工程 | 1 | 77 |  | 20191060002 | 李四 | 女 | 通信工程 | 2 | 80 |  | 20191060002 | 李四 | 女 | 通信工程 | 4 | 85 |  | 20191060003 | 王五 | 男 | 计算机 | 1 | 90 |  | 20191060003 | 王五 | 男 | 计算机 | 2 | 92 |  | 20191060003 | 王五 | 男 | 计算机 | 3 | 90 |  | 20191060003 | 王五 | 男 | 计算机 | 5 | 91 |  | 20191060004 | 赵六 | 男 | 通信工程 | 1 | 88 |  | 20191060004 | 赵六 | 男 | 通信工程 | 6 | 90 |  | 20191060005 | 钱七 | 女 | 计算机 | 1 | 86 |  | 20191060005 | 钱七 | 女 | 计算机 | 5 | 88 |  +-------------+--------+------+--------------+------+-------+  14 rows in set (0.00 sec) |

应注意，在表达外连接时SQL有其关键字OUTER JOIN，而一般的连接直接用AND隐式表达了。同时在选择属性时，由于sno在student表和sc表中都存在，所以需要指定表，而其他属性都为某个表独有，所以不需指定。

（在2、3 题中，由于该库的数据量非常大，所以对于结果数大于20的表均使用“LIMIT 20”约束，仅显示前20条）

2.

(1) 简单的单表查询：查询不同语言的使用情况

|  |  |
| --- | --- |
| SELECT      DISTINCT count(\*),      Language  FROM countrylanguage  GROUP BY(Language)  ORDER BY count(\*) DESC  LIMIT 50; | mysql> SELECT  -> DISTINCT count(\*),  -> Language  -> FROM countrylanguage  -> GROUP BY(Language)  -> ORDER BY count(\*) DESC  -> LIMIT 20;  +----------+----------------+  | count(\*) | Language |  +----------+----------------+  | 60 | English |  | 33 | Arabic |  | 28 | Spanish |  | 25 | French |  | 19 | Chinese |  | 19 | German |  | 17 | Russian |  | 15 | Italian |  | 14 | Creole English |  | 12 | Ukrainian |  | 12 | Ful |  | 12 | Portuguese |  | 12 | Turkish |  | 10 | Polish |  | 9 | Serbo-Croatian |  | 9 | Creole French |  | 8 | Romani |  | 8 | Hungarian |  | 8 | Hindi |  | 7 | Belorussian |  +----------+----------------+  20 rows in set (0.00 sec) |

果然，英语使用的区域还是最广的。

(2) ORDER BY的单表查询：查询各个国家的人数，并降序排列

|  |  |
| --- | --- |
| SELECT Name, Population  FROM country  ORDER BY Population DESC  LIMIT 20; | mysql> SELECT Name, Population  -> FROM country  -> ORDER BY Population DESC  -> LIMIT 20;  +--------------------+------------+  | Name | Population |  +--------------------+------------+  | China | 1277558000 |  | India | 1013662000 |  | United States | 278357000 |  | Indonesia | 212107000 |  | Brazil | 170115000 |  | Pakistan | 156483000 |  | Russian Federation | 146934000 |  | Bangladesh | 129155000 |  | Japan | 126714000 |  | Nigeria | 111506000 |  | Mexico | 98881000 |  | Germany | 82164700 |  | Vietnam | 79832000 |  | Philippines | 75967000 |  | Egypt | 68470000 |  | Iran | 67702000 |  | Turkey | 66591000 |  | Ethiopia | 62565000 |  | Thailand | 61399000 |  | United Kingdom | 59623400 |  +--------------------+------------+  20 rows in set (0.00 sec) |

不出意外的，中国人数占世界第1（不过这个数据可能比较旧了）

(3) GROUP BY+HAVING+ORDER BY的单表查询：查询在city中的个数大于等于5个的国家，按降序排列

|  |  |
| --- | --- |
| SELECT      CountryCode,      COUNT(\*) AS '城市数'  FROM city  GROUP BY(CountryCode)  HAVING COUNT(\*) >= 5  ORDER BY COUNT(\*) DESC  LIMIT 20; | mysql> SELECT  -> CountryCode,  -> COUNT(\*) AS '城市数'  -> FROM city  -> GROUP BY(CountryCode)  -> HAVING COUNT(\*) >= 5  -> ORDER BY COUNT(\*) DESC  -> LIMIT 20;  +-------------+-----------+  | CountryCode | 城市数 |  +-------------+-----------+  | CHN | 363 |  | IND | 341 |  | USA | 274 |  | BRA | 250 |  | JPN | 248 |  | RUS | 189 |  | MEX | 173 |  | PHL | 136 |  | DEU | 93 |  | IDN | 85 |  | GBR | 81 |  | KOR | 70 |  | IRN | 67 |  | NGA | 64 |  | TUR | 62 |  | PAK | 59 |  | ESP | 59 |  | ITA | 58 |  | ARG | 57 |  | UKR | 57 |  +-------------+-----------+  20 rows in set (0.00 sec) |

(4) 选择表中的若干元组（WHERE IN）：查询五常国家的城市

|  |  |
| --- | --- |
| SELECT Name  FROM city  WHERE      CountryCode IN (          'CHN',          'RUS',          'USA',          'GBR',          'FRA'      )  LIMIT 20; | mysql> SELECT Name  -> FROM city  -> WHERE  -> CountryCode IN (  -> 'CHN',  -> 'RUS',  -> 'USA',  -> 'GBR',  -> 'FRA'  -> )  -> LIMIT 20;  +--------------------+  | Name |  +--------------------+  | Shanghai |  | Peking |  | Chongqing |  | Tianjin |  | Wuhan |  | Harbin |  | Shenyang |  | Kanton [Guangzhou] |  | Chengdu |  | Nanking [Nanjing] |  | Changchun |  | Xi´an |  | Dalian |  | Qingdao |  | Jinan |  | Hangzhou |  | Zhengzhou |  | Shijiazhuang |  | Taiyuan |  | Kunming |  +--------------------+  20 rows in set (0.00 sec) |

由于city表中，顺序根据CountryCode升序排列，中国代号是CHN，排在最前面，所以前20个就都是中国的城市。

(5) 选择表中的若干元组：查询人口在1亿到10亿的国家

|  |
| --- |
| SELECT Name, Population  FROM country  WHERE      Population BETWEEN 100000000 AND 1000000000; |
| mysql> SELECT Name, Population  -> FROM country  -> WHERE  -> Population BETWEEN 100000000 AND 1000000000;  +--------------------+------------+  | Name | Population |  +--------------------+------------+  | Bangladesh | 129155000 |  | Brazil | 170115000 |  | Indonesia | 212107000 |  | Japan | 126714000 |  | Nigeria | 111506000 |  | Pakistan | 156483000 |  | Russian Federation | 146934000 |  | United States | 278357000 |  +--------------------+------------+  8 rows in set (0.00 sec) |

**(6) 包含聚集函数的单表查询：查询各大洲的国家数**

|  |  |
| --- | --- |
| SELECT      Continent,      COUNT(\*) AS '国家数'  FROM country  GROUP BY(Continent)  ORDER BY(COUNT(\*)) DESC; | mysql> SELECT  -> Continent,  -> COUNT(\*) AS '国家数'  -> FROM country  -> GROUP BY(Continent)  -> ORDER BY(COUNT(\*)) DESC;  +---------------+-----------+  | Continent | 国家数 |  +---------------+-----------+  | Africa | 58 |  | Asia | 51 |  | Europe | 46 |  | North America | 37 |  | Oceania | 28 |  | South America | 14 |  | Antarctica | 5 |  +---------------+-----------+  7 rows in set (0.00 sec) |

3.

(1) 查询人数最多的城市所在的大洲

|  |
| --- |
| SELECT city.Name, country.Continent  FROM city, country  WHERE      city.CountryCode = country.Code      AND city.Population = (SELECT MAX(Population) FROM city); |
| mysql> SELECT city.Name, country.Continent  -> FROM city, country  -> WHERE  -> city.CountryCode = country.Code  -> AND city.Population = (SELECT MAX(Population) FROM city);  +-----------------+-----------+  | Name | Continent |  +-----------------+-----------+  | Mumbai (Bombay) | Asia |  +-----------------+-----------+  1 row in set (0.00 sec) |

(2) 简单的连接查询：查询亚洲国家及其官方语言

|  |  |
| --- | --- |
| SELECT      country.Name AS '国家',      countrylanguage.Language AS '官方语言'  FROM country, countrylanguage  WHERE      country.Code = countrylanguage.CountryCode      AND country.Continent = 'Asia'  AND countrylanguage.IsOfficial = 'T'  LIMIT 20; | mysql> SELECT  -> country.Name AS '国家',  -> countrylanguage.Language AS '官方语言'  -> FROM country, countrylanguage  -> WHERE  -> country.Code = countrylanguage.CountryCode  -> AND country.Continent = 'Asia'  -> AND countrylanguage.IsOfficial = 'T'  -> LIMIT 20;  +----------------------+--------------+  | 国家 | 官方语言 |  +----------------------+--------------+  | Afghanistan | Dari |  | Afghanistan | Pashto |  | United Arab Emirates | Arabic |  | Armenia | Armenian |  | Azerbaijan | Azerbaijani |  | Bangladesh | Bengali |  | Bahrain | Arabic |  | Brunei | Malay |  | Bhutan | Dzongkha |  | China | Chinese |  | Cyprus | Greek |  | Cyprus | Turkish |  | Georgia | Georgiana |  | Hong Kong | English |  | Indonesia | Malay |  | India | Hindi |  | Iran | Persian |  | Iraq | Arabic |  | Israel | Arabic |  | Israel | Hebrew |  +----------------------+--------------+  20 rows in set (0.00 sec) |

(3) 自身连接：查询使用中国使用的语言的国家

|  |  |
| --- | --- |
| SELECT      OTHER.CountryCode AS '国家',      OTHER.Language AS '语言',      CHN.Language AS '中国使用的语言'  FROM      countrylanguage OTHER,      countrylanguage CHN  WHERE      OTHER.Language = CHN.Language      AND OTHER.CountryCode != 'CHN'      AND CHN.CountryCode = 'CHN'  LIMIT 20; | mysql> SELECT  -> OTHER.CountryCode AS '国家',  -> OTHER.Language AS '语言',  -> CHN.Language AS '中国使用的语言'  -> FROM  -> countrylanguage OTHER,  -> countrylanguage CHN  -> WHERE  -> OTHER.Language = CHN.Language  -> AND OTHER.CountryCode != 'CHN'  -> AND CHN.CountryCode = 'CHN'  -> LIMIT 20;  +--------+-----------+-----------------------+  | 国家 | 语言 | 中国使用的语言 |  +--------+-----------+-----------------------+  | BRN | Chinese | Chinese |  | CAN | Chinese | Chinese |  | CRI | Chinese | Chinese |  | CXR | Chinese | Chinese |  | JPN | Chinese | Chinese |  | KHM | Chinese | Chinese |  | KOR | Chinese | Chinese |  | MNG | Mongolian | Mongolian |  | MNP | Chinese | Chinese |  | MYS | Chinese | Chinese |  | NRU | Chinese | Chinese |  | PLW | Chinese | Chinese |  | PRK | Chinese | Chinese |  | PYF | Chinese | Chinese |  | REU | Chinese | Chinese |  | SGP | Chinese | Chinese |  | THA | Chinese | Chinese |  | USA | Chinese | Chinese |  | VNM | Chinese | Chinese |  | VNM | Miao | Miao |  +--------+-----------+-----------------------+  20 rows in set (0.00 sec) |

(4) 多表连接：查询各大洲的语言数

|  |  |
| --- | --- |
| SELECT      DISTINCT Continent,      COUNT(Language)  FROM      country,      city,      countrylanguage  WHERE      country.Code = countrylanguage.CountryCode      AND country.Code = city.CountryCode  GROUP BY(Continent); | mysql> SELECT  -> DISTINCT Continent,  -> COUNT(Language)  -> FROM  -> country,  -> city,  -> countrylanguage  -> WHERE  -> country.Code = countrylanguage.CountryCode  -> AND country.Code = city.CountryCode  -> GROUP BY(Continent);  +---------------+-----------------+  | Continent | COUNT(Language) |  +---------------+-----------------+  | North America | 5106 |  | Asia | 15169 |  | Africa | 2522 |  | Europe | 5639 |  | South America | 2007 |  | Oceania | 227 |  +---------------+-----------------+  6 rows in set (0.01 sec) |

(5) 多表连接：查询名字最长的城市及所在国家和大洲

|  |
| --- |
| SELECT      city.Name AS '城市名', country.Name, country.Continent  FROM city, country  WHERE      city.CountryCode = country.Code      AND LENGTH(city.Name) = (SELECT MAX(LENGTH(Name)) FROM city); |
| mysql> SELECT  -> city.Name AS '城市名', country.Name, country.Continent  -> FROM city, country  -> WHERE  -> city.CountryCode = country.Code  -> AND LENGTH(city.Name) = (SELECT MAX(LENGTH(Name)) FROM city);  +------------------------------------+------------+-----------+  | 城市名 | Name | Continent |  +------------------------------------+------------+-----------+  | Luxembourg [Luxemburg/Lëtzebuerg] | Luxembourg | Europe |  +------------------------------------+------------+-----------+  1 row in set (0.00 sec) |

四、思考题

1.

WHERE是对FROM的数据表进行数据限制，而HAVING是对聚集函数的结果进行数据限制。