

数据库系统原理实验一、实验二、实验三

实验报告

毛子恒

2019211397

北京邮电大学 计算机学院

日期：2021 年 12 月 16 日

Part I

实验一

1 概述

1.1 实验目的

1. 通过对 GaussDB(for openGauss) 数据库创建与访问：
 - (a) 了解华为云分布式数据库 GaussDB(for openGauss) 的软件环境和创建方法；
 - (b) 掌握并熟悉 GaussDB(for openGauss) 数据库软件的使用方法；
 - (c) 掌握并熟悉 GaussDB(for openGauss) 数据库软件的构成和相关工具；
 - (d) 通过 GaussDB(for openGauss) 数据库软件的使用，深入理解数据库系统的基本概念。
2. 通过创建 GaussDB(for openGauss) 数据库及进行相应的维护，了解并掌握 GaussDB(for openGauss) 数据库的创建与维护的不同方法和途径，进而通过这一具体的数据库理解实际数据库所包含的各要素。

1.2 实验平台及环境

- 华为云 GaussDB(for openGauss) 1.4.1
- 数据库兼容：PostgreSQL

1.3 实验内容

1. GaussDB(for openGauss) 数据库软件的使用：
 - (a) 登录并运行 GaussDB(for openGauss) 数据库；
 - (b) 了解华为云数据库 GaussDB(for openGauss) 的机制；
 - (c) 熟悉 GaussDB(for openGauss) 数据库的各项功能。
2. 数据库创建与维护：

- (a) 创建“疫情数据”数据库；
- (b) 对数据库属性和参数进行相应的修改和维护；
- (c) 练习数据库的删除等维护操作。

1.4 实验步骤

1. 熟悉 GaussDB(for openGauss) 数据库的创建过程。
2. 创建一个名为“疫情数据”的数据库；
3. 删除“疫情数据”数据库。

2 实验结果及分析

数据库的登录

1. 登录华为云控制台，进入数据库管理服务器 DAS；
2. 进入开发工具；
3. 新增数据库实例登录，数据库引擎设置为 GaussDB(for openGauss)，设置用户名和密码；
4. 登录刚才创建的数据库实例。

数据库的创建 新建数据库，命名为"Epidemic_data2019211397"，如图 1。




图 1: 创建数据库

对应 SQL 语句为：

```
1 CREATE DATABASE Epidemic_data2019211397;
```

数据库的删除 点击“删除库”，如图 2。

Epidemic_data2019211397 UTF8 库管理 | SQL查询 | 删除库

图 2: 删除数据库

对应 SQL 语句为:

```
1 DROP DATABASE Epidemic_data2019211397;
```

3 实验总结

本次实验中我基本熟悉了华为云数据库的使用，在 IAM 账号下进行了数据库的创建和删除操作，期间依照实验步骤进行操作，没有问题产生。

Part II

实验二

4 概述

4.1 实验目的

1. 通过进行数据库表的建立操作，熟悉并掌握 GaussDB(for openGauss) 数据库表的建立方法，理解关系型数据库表的结构，巩固 PostgreSQL 中关于数据库表的建立语句；
2. 通过进行数据库表数据的增加、删除和插入等维护操作，熟悉并掌握 GaussDB(for openGauss) 数据库数据的操作方法，巩固 PostgreSQL 中关于数据维护的语句。

4.2 实验平台及环境

- 华为云 GaussDB(for openGauss) 1.4.1
- 数据库兼容：PostgreSQL

4.3 实验内容

建立相应的表并熟悉基本操作，例如建表、对表进行增、删、改、查。

4.4 实验步骤

1. 熟悉课程实验背景知识；
2. 使用 GaussDB(for openGauss) 数据库软件创建相应的表；
3. 将提供的数据导入各表，掌握 GaussDB(for openGauss) 数据库数据导入的方法；注意：
 - (a) 表中空列的处理；
 - (b) 表结构与数据类型的匹配。
4. 修改“病例基本信息”表数据，增加名为“备注”的列，数据类型为 `varchar()` 型。
5. 修改“病例基本信息”表数据，将“备注”列的数据类型改为 `int`。

6. 修改“病例基本信息”表数据，删除“备注”列。
7. 删除“病例基本信息”数据表。

5 实验结果及分析

数据表的创建 观察数据表的结构，执行如下 SQL 语句来创建表：

```
1 CREATE TABLE 病例行程信息表 (  
2     行程号 INTEGER NULL,  
3     病例号 INTEGER NULL,  
4     日期信息 VARCHAR(100) NULL,  
5     行程信息 VARCHAR(250) NULL  
6 );  
7 CREATE TABLE 美国各州县确诊与死亡数统计表 (  
8     日期 DATE NULL,  
9     国家 VARCHAR(50) NULL,  
10    州 VARCHAR(50) NULL,  
11    县 VARCHAR(50) NULL,  
12    累计确诊 INTEGER NULL,  
13    累计死亡 INTEGER NULL  
14 );  
15 CREATE TABLE 全国各省累计数据统计表 (  
16    日期 DATE NULL,  
17    省 VARCHAR(50) NULL,  
18    累计确诊 INTEGER NULL,  
19    累计治愈 INTEGER NULL,  
20    累计死亡 INTEGER NULL  
21 );  
22 CREATE TABLE 病例基本信息表 (  
23    病例号 INTEGER NULL,  
24    省 VARCHAR(50) NULL,  
25    市 VARCHAR(50) NULL,  
26    区 VARCHAR(50) NULL,  
27    日期 DATE NULL,  
28    性别 VARCHAR(50) NULL,  
29    年龄 INTEGER NULL,  
30    患者信息 VARCHAR(100) NULL,  
31    其他信息 VARCHAR(200) NULL,  
32    信息来源 VARCHAR(50) NULL  
33 );  
34 CREATE TABLE 参考信息表 (  
35    组合码 VARCHAR(100) NULL,  
36    国家地区 VARCHAR(50) NULL,  
37    省州 VARCHAR(50) NULL,  
38    市县 VARCHAR(50) NULL,  
39    纬度 DOUBLE PRECISION NULL,  
40    经度 DOUBLE PRECISION NULL,  
41    人口数 INTEGER NULL  
42 );
```

```

43 CREATE TABLE 各国疫情数据统计表 (
44     日期 DATE NULL,
45     国家地区 VARCHAR(50) NULL,
46     省州 VARCHAR(50) NULL,
47     累计确诊 INTEGER NULL,
48     累计治愈 INTEGER NULL,
49     累计死亡 INTEGER NULL
50 );
51 CREATE TABLE 全国城市风险等级表 (
52     省 VARCHAR(50) NULL,
53     市 VARCHAR(50) NULL,
54     区 VARCHAR(50) NULL,
55     地址详情 VARCHAR(100) NULL,
56     风险等级 VARCHAR(50) NULL
57 );
58 CREATE TABLE 全国各省参考信息表 (
59     中文名称 VARCHAR(50) NULL,
60     英文名称 VARCHAR(50) NULL,
61     组合码 VARCHAR(50) NULL,
62     纬度 DOUBLE PRECISION NULL,
63     经度 DOUBLE PRECISION NULL,
64     人口数 INTEGER NULL
65 );

```

导入各表 点击数据管理服务中的“导入”菜单，点击“新建任务”，选择 CSV 附件，上传文件，选择对应的数据库和表，执行导入任务，如图 3。

新建任务

导入类型: sql, **CSV**

文件来源: **上传文件**, 从OBS中选择

附件存放位置: obs-1ebf 没有OBS桶? 创建OBS桶
创建OBS桶免费, 但保存文件将产生一定的费用。

选择附件:
 +
 点击或将文件拖动到此处后上传文件 (.csv)
 全国各省参考信息表_1635752506622.csv
 最大不能超过1GB, 且只能上传一个附件

数据库: bupt2019211397

表: "root". "全国各省参考信息表"

数据位置: **第1行为属性**, 第1行为数据

字符集: **自动检测**, UTF8, GBK

选项:
☐ 忽略报错, 即SQL执行失败时跳过
☒ 导入完成后删除上传的文件
☐ 清空表, 执行导入前先对相应的表执行Truncate操作

备注:

创建导入任务 取消

图 3: 导入表

对于存在空列（集包括列头的整列都为空）的表，利用 Excel 将空列删除即可。

修改“病例基本信息”表数据 增加名为“备注”的列，数据类型为varchar()。

```
1 ALTER TABLE 病例基本信息表 ADD 备注 varchar(100);
```

将“备注”列的数据类型改为int。

```
1 ALTER TABLE 病例基本信息表 ALTER COLUMN 备注 TYPE INTEGER;
```

删除“备注”列。

```
1 ALTER TABLE 病例基本信息表 DROP COLUMN 备注;
```

删除“病例基本信息”数据表 SQL 语句如下：

```
1 DROP TABLE 病例基本信息表;
```

6 实验总结

在导入数据的过程中，出现导入错误的情况，经查看错误日志，大部分情况都是varchar的长度限制太小所致，增大长度限制之后即可正常导入。

在导入过程中注意细分列的类型，比如病例形成信息表的形成号和病历号设为 int 类型，日期列都设为 date 类型，这样有助于后续的查询操作。

实验使我对基本的 SQL 建表和修改表的操作更加熟悉，实验过程中没有遇到其它问题。

Part III

实验三

7 概述

7.1 实验目的

通过对实验二建立的数据库关系表的各种查询的操作，加深对 SQL 语言和 PostgreSQL 查询语言的了解，掌握相关查询语句的语法及使用方法。

7.2 实验平台及环境

- 华为云 GaussDB(for openGauss) 1.4.1
- 数据库兼容：PostgreSQL

7.3 实验内容

7.3.1 单表查询

1. 查询国内确诊病例基本信息的所有信息来源；
2. 给出河南省、西藏自治区、台湾省的英文名称和人口数；
3. 查询 2021 年 1 月 20 日各省现有确诊病例数据，按现有确诊病例数降序排列输出。计算方法：现有确诊数 = 累计确诊 - 累计死亡 - 累计治愈；
4. 计算截至 2021 年 1 月 20 日全国累计确诊病例数；
5. 查询 1005 号病例确诊后，其所在市新增的所有确诊病例；
6. 在“病例基本信息表”中查询石家庄市在 2021 年 1 月 11 日当天以及之前的所有 60 岁以上的患者信息；
7. 统计截止到 2020 年 12 月 30 日美国累计确诊病例数最多的 10 个州；
8. 统计截止到 2020 年 12 月 30 日美国新冠疫情死亡率高于 2% 的州。

7.3.2 多表查询

1. 借助病例行程信息粗略查询曾去过“源升品质生活坊”的所有患者的基本信息；
2. 根据病例行程信息表和病例基本信息表，查询行程信息中存在“家庭聚餐”的病例被确诊的日期；
3. 对比中美两国累计确诊病例数，输出格式为(日期，中国累计确诊，美国累计确诊)；
4. 计算截止到 2021 年 1 月 20 日，美国有些县的累计确诊是同一个州的其他县的 2 倍或以上，列出这些县，以及他们所在的州和他们的累计确诊；
5. 计算世界上人口数排名前 10 位的国家地区；
6. 列出美国人口超千万的大州中，截至 2021 年 1 月 20 日新冠肺炎疫情死亡率超过 2% 的州；
7. 截至 2021 年 1 月 20 日，河北省哪些区出现了新冠确诊病例但不属于中高风险地区；
8. 在病例行程信息表的基础上根据病例基本信息表，查询河北省病例的信息：包括行程 ID，病例 ID，患者信息，日期信息，行程信息。

7.3.3 嵌套查询

1. 查询披露的确诊患者信息中年龄最大的患者，输出其基本信息。(未注明年龄的患者不进行比较)；
2. 查询 2020 年 12 月份新增确诊患者最多的城市；
3. 结合“全国各省参考信息表”和“病例基本信息表”给出没有新增确诊病例或未披露病例信息的省份；

4. 2021 年 1 月 20 日全国中高风险地区所在省中，哪些省在 1 月 20 日没有新增确诊信息披露；
5. 根据病例基本信息表查询一月份国内新增患者病例最多的城市；
6. 查询除中美两国以外的其余国家中，进入 2021 年以来单日新增确诊病例始终不低于一万个的国家。

8 实验结果及分析

国内确诊病例基本信息的所有信息来源 SQL 语句如下：

- 1 `SELECT DISTINCT` 信息来源
- 2 `FROM` 病例基本信息表；

语句从病例基本信息表中选出不重复的信息来源。查询结果如图 4。

	信息来源
1	新京报
2	澎湃新闻
3	青冈县疾病预防控制中心
4	大众网
5	今晚报
6	中国新闻网
7	广西新闻网
8	辽宁日报
9	黑龙江卫健委网站
10	河北日报
11	黑龙江省齐齐哈尔市疾控中心

图 4: 查询结果 1

河南省、西藏自治区、台湾省的英文名称和人口数 SQL 语句如下：

- 1 `SELECT` 英文名称，人口数
- 2 `FROM` 全国各省参考信息表
- 3 `WHERE` 中文名称 `IN` ('河南省', '西藏自治区', '台湾省');

语句从各省参考信息表中选出三个省的元组，再选出英文名称和人口数。查询结果如图 5。

	英文名称	人口数
1	Henan	96850000
2	Taiwan	23816775
3	Tibet	3440000

图 5: 查询结果 2

2021 年 1 月 20 日各省现有确诊病例数据，按现有确诊病例数降序排列输出 SQL 语句如下：

- 1 `SELECT` 省，累计确诊-累计死亡-累计治愈 `AS` 现有确诊数
- 2 `FROM` 全国各省累计数据统计表


```

3 WHERE 日期 = '2021-01-20'
4 ORDER BY 现有确诊数 DESC;

```

语句查询该日各省的数据，并按现有确诊数降序输出。查询结果如图 6。

	省	现有确诊数
1	河北省	829
2	香港特别行政区	766
3	黑龙江省	259
4	吉林省	162
5	上海市	95
6	北京市	44
7	广东省	39
8	陕西省	26
9	辽宁省	25
10	天津市	24
11	福建省	19

图 6: 查询结果 3

截至 2021 年 1 月 20 日全国累计确诊病例数 SQL 语句如下：

```

1 SELECT SUM(累计确诊) AS 全国累计确诊病例数
2 FROM 全国各省累计数据统计表
3 WHERE 日期 = '2021-01-20';

```

语句从累计数据统计表中查询当日各省的累计确诊数之和。查询结果如图 7。

	全国累计确诊病例数
1	98545

图 7: 查询结果 4

1005 号病例确诊后，其所在市新增的所有确诊病例 SQL 语句如下：

```

1 SELECT *
2 FROM 病例基本信息表 AS T1
3 WHERE EXISTS (
4     SELECT 日期, 市
5     FROM 病例基本信息表 AS T2
6     WHERE T2.病例号 = 1005 AND T1.日期 >= T2.日期 AND T1.市 = T2.市
7 );

```

或

```

1 WITH 病例1005 AS (
2     SELECT 日期, 市
3     FROM 病例基本信息表
4     WHERE 病例号 = 1005

```

```

5 )
6 SELECT 病例基本信息表.*
7 FROM 病例基本信息表, 病例1005
8 WHERE 病例基本信息表.日期 >= 病例1005.日期 AND 病例基本信息表.市 = 病例1005.市;

```

语句利用相关子查询，从WHERE子句中筛选出病例 1005 的关系和外层的关系比较；或者可以采用WITH子句创建一个临时关系，用来表示病例 1005，之后再从病例基本信息表中选取元组和这个关系进行比较。查询结果如图 8。

	病例号	省	市	区	日期	性别	年龄	患者信息
1	982	黑龙江省	黑河市	爱辉区	2021-01-07	女	54	女, 54岁
2	1004	黑龙江省	黑河市	爱辉区	2021-01-06	女	41	女, 41岁
3	1005	黑龙江省	黑河市	爱辉区	2021-01-06	女	18	无症状感染

图 8: 查询结果 5

石家庄市在 2021 年 1 月 11 日当天以及之前的所有 60 岁以上的患者信息 SQL 语句如下：

```

1 SELECT *
2 FROM 病例基本信息表
3 WHERE 日期 <= '2021-01-11' AND 市 = '石家庄市' AND 年龄 >= 60;

```

语句按照给定条件筛选元组。查询到 90 条结果（如果条件设为年龄 > 60 则查询到 84 条结果）查询结果如图 9。

	病例号	省	市	区	日期	性别	年龄	患者信息
1	925	河北省	石家庄市	井陉县	2021-01-08	女	65	女, 65岁, 井
2	741	河北省	石家庄市	藁城区	2021-01-11	女	78	确诊病例6:
3	743	河北省	石家庄市	藁城区	2021-01-11	女	75	确诊病例8:
4	744	河北省	石家庄市	藁城区	2021-01-11	女	68	确诊病例9:
5	746	河北省	石家庄市	藁城区	2021-01-11	女	77	确诊病例11:
6	747	河北省	石家庄市	藁城区	2021-01-11	男	70	确诊病例12:
7	751	河北省	石家庄市	藁城区	2021-01-11	女	67	确诊病例16:
8	755	河北省	石家庄市	藁城区	2021-01-11	女	68	确诊病例20:

图 9: 查询结果 6

截止到 2020 年 12 月 30 日美国累计确诊病例数最多的 10 个州 SQL 语句如下：

```

1 SELECT 州, SUM(累计确诊) AS 州累计确诊
2 FROM 美国各州县确诊与死亡数统计表
3 WHERE 日期 = '2020-12-30'
4 GROUP BY 州
5 ORDER BY 州累计确诊 DESC
6 LIMIT 10;

```

语句筛选出当日的的数据，并且按照州分组，通过累计确诊降序排列，再筛选出前十个。查询结果如图 10。

	州	州累计确诊
1	California	2291414
2	Texas	1754975
3	Florida	1386123
4	New York	963398
5	Illinois	955388
6	Ohio	698748
7	Georgia	654743
8	Pennsylvania	636861
9	Tennessee	588889
10	North Carolina	532838

图 10: 查询结果 7

截止到 2020 年 12 月 30 日美国新冠疫情死亡率高于 2% 的州 SQL 语句如下:

```

1 SELECT 州, 州累计死亡/州累计确诊 AS 死亡率
2 FROM (
3     SELECT 州, SUM(累计死亡) AS 州累计死亡, SUM(累计确诊) AS 州累计确诊
4     FROM 美国各州县确诊与死亡数统计表
5     WHERE 日期 = '2020-12-30'
6     GROUP BY 州
7 )
8 WHERE 州累计确诊 <> 0 AND 死亡率 > 0.02;
```

语句首先在 FROM 子查询中筛选出当日的的数据, 按照各州的累计死亡和累计确诊聚合, 最后在外层筛选出死亡率符合条件的州。查询结果如图 11。

	州	死亡率
1	Connecticut	.0324725176081698784
2	Massachusetts	.0335346463878575543
3	Pennsylvania	.024628141811632531
4	Michigan	.0246263390973873522
5	Rhode Island	.0282048914711935312
6	Louisiana	.0239389318861674197
7	Grand Princess	.0291262135922398187
8	Mississippi	.0222886317617516684
9	Maryland	.0213673183796286645
10	District of Columbia	.0271228875443354882
11	Ohio	.0287456264882793587

图 11: 查询结果 8

曾去过“源升品质生活坊”的所有患者的基本信息 SQL 语句如下:

```

1 SELECT 病例基本信息表.*
2 FROM 病例基本信息表 NATURAL JOIN 病例行程信息表
3 WHERE 行程信息 LIKE '% 源升品质生活坊%';
```

语句从病例基本信息表和病例行程信息表的自然连接（按照病例 ID 连接），筛选出行程信息中含有子串“源升品质生活坊”的元组。查询结果如图 12。

行程信息中存在“家庭聚餐”的病例被确诊的日期 SQL 语句如下:

	病例号	省	市	区	日期	性别	年龄	患者信息
1	157	吉林省	通化市	东昌区	2021-01-18	女	73	确诊病例21
2	167	吉林省	通化市	东昌区	2021-01-18	女	76	无症状感染者
3	171	吉林省	通化市	东昌区	2021-01-18	男	78	无症状感染者
4	568	吉林省	通化市	东昌区	2021-01-13	男	45	无症状感染者

图 12: 查询结果 9

```

1 SELECT 日期
2 FROM 病例基本信息表 NATURAL JOIN 病例行程信息表
3 WHERE 行程信息 LIKE '% 家庭聚餐%';

```

与上个查询类似。查询结果如图 13。

	日期
1	2021-01-13
2	2021-01-13
3	2021-01-05
4	2021-01-04

图 13: 查询结果 10

对比中美两国累计确诊病例数 SQL 语句如下：

```

1 SELECT 日期, 中国累计确诊, 美国累计确诊
2 FROM (
3     SELECT 日期, SUM(累计确诊) AS 中国累计确诊
4     FROM 全国各省累计数据统计表
5     GROUP BY 日期
6 ) AS 中国确诊
7 NATURAL JOIN
8 (
9     SELECT 日期, SUM(累计确诊) AS 美国累计确诊
10    FROM 美国各州县确诊与死亡数统计表
11    GROUP BY 日期
12 ) AS 美国确诊

```

语句查询中国的累计确诊信息和美国的累计确诊信息，将两个表按日期自然连接。查询结果如图 14。

截止到 2021 年 1 月 20 日，美国有些县的累计确诊是同一个州的其他县的 2 倍或以上 SQL 语句如下：

```

1 WITH 单日数据 AS (
2     SELECT 州, 县, 累计确诊
3     FROM 美国各州县确诊与死亡数统计表
4     WHERE 日期 = '2021-01-20'

```

	日期	中国累计确诊	美国累计确诊
1	2021-01-09	96691	22225728
2	2020-12-13	94329	16488428
3	2020-12-26	95542	19067574
4	2021-01-05	96311	21140217
5	2020-11-22	92117	12347905
6	2020-11-29	92815	13495104
7	2021-01-19	98324	24281531
8	2020-12-22	95229	18321157
9	2020-12-24	95384	18743916
10	2021-01-17	97948	23962788
11	2020-11-24	92297	12697001

图 14: 查询结果 11

```

5 )
6 SELECT 县, 州, 州累计确诊
7 FROM (
8     SELECT 州, SUM(累计确诊) AS 州累计确诊
9     FROM 单日数据
10    GROUP BY 州
11 ) AS 州确诊
12 NATURAL JOIN
13 (
14     SELECT 州, 县
15     FROM 单日数据 AS T1
16     WHERE 累计确诊 >= SOME (
17         SELECT 累计确诊 * 2
18         FROM 单日数据 AS T2
19         WHERE T1.州 = T2.州
20     )
21 ) AS 确诊多的县;

```

语句首先筛选出当日的各州县的累计确诊数据作为临时关系，之后利用相关子查询，筛选出确诊数比同一个州的某些县多两倍的县，将这些县与各州的总确诊数自然连接。查询结果如图 15。

	县	州	州累计确诊
1	Saline	Illinois	1081354
2	Seminole	Florida	1601011
3	Alpine	California	3097632
4	Creek	Oklahoma	360360
5	Fillmore	Minnesota	449492
6	Towns	Georgia	836649
7	Taylor	Wisconsin	573119
8	Peach	Georgia	836649
9	Custer	Oklahoma	360360
10	Sullivan	Indiana	598313
11	Waller	Texas	2185554

图 15: 查询结果 12

世界上人口数排名前 10 位的国家地区 SQL 语句如下：

```

1 SELECT *
2 FROM (

```

```

3      SELECT 国家地区, 人口数
4      FROM 参考信息表
5      WHERE 人口数 IS NOT NULL AND 组合码 = 国家地区
6  )
7  UNION
8  (
9      SELECT 组合码 AS 国家地区, 人口数
10     FROM 全国各省参考信息表
11     WHERE 组合码 = 'China'
12  )
13 ORDER BY 人口数 DESC
14 LIMIT 10;

```

语句提取出参考信息表中的各个国家的总人口信息，再并上中国的总人口数。查询结果如图 16。

	国家地区	人口数
1	China	1404676330
2	India	1380004385
3	US	329466283
4	Indonesia	273523621
5	Pakistan	220892331
6	Brazil	212559409
7	Nigeria	206139587
8	Bangladesh	164689383

图 16: 查询结果 13

美国人口超千万的大州中，截至 2021 年 1 月 20 日新冠肺炎疫情死亡率超过 2% 的州 SQL 语句如下：

```

1  SELECT 州, 州累计死亡/州累计确诊 AS 死亡率
2  FROM (
3      SELECT 州, SUM(累计死亡) AS 州累计死亡, SUM(累计确诊) AS 州累计确诊
4      FROM 美国各州县确诊与死亡数统计表 AS T1
5      WHERE 日期 = '2021-01-20'
6      GROUP BY 州
7      HAVING 10000000 < (
8          SELECT 人口数
9          FROM 参考信息表 AS T2
10         WHERE T1.州 = T2.省州 AND T2.国家地区 = 'US' AND T2.市县 IS NULL
11     )
12 )
13 WHERE 州累计确诊 <> 0 AND 死亡率 > 0.02;

```

语句基于之前的查询，再在 FROM 子查询中添加人口数超过千万的限制，这一限制通过相关子查询实现。查询结果如图 17。

截至 2021 年 1 月 20 日，河北省出现了新冠确诊病例但不属于中高风险的地区 SQL 语句如下：

	州	死亡率
1	Pennsylvania	.0251541422883146439
2	New York	.032494641786951356

图 17: 查询结果 14

```

1 SELECT DISTINCT 区
2 FROM 病例基本信息表
3 WHERE 日期 < '2021-01-20' AND 省 = '河北省' AND 区 IS NOT NULL
4 EXCEPT
5 SELECT DISTINCT 区
6 FROM 全国城市风险等级表
7 WHERE 省 = '河北省'

```

或

```

1 SELECT DISTINCT 区
2 FROM 病例基本信息表
3 WHERE 日期 < '2021-01-20' AND 省 = '河北省' AND 区 IS NOT NULL AND 区 NOT IN (
4     SELECT DISTINCT 区
5     FROM 全国城市风险等级表
6     WHERE 省 = '河北省'
7 );

```

语句用出现确诊病例的区的集合减去中高风险地区集合，可以直接用EXCEPT实现，也可以用NOT IN实现。查询结果如图 18。

	区
1	荣成
2	井陉县
3	裕华市
4	正定

图 18: 查询结果 15

在病例行程信息表的基础上根据病例基本信息表，查询河北省病例的信息 SQL 语句如下：

```

1 SELECT 行程号，病例号，患者信息，日期信息，行程信息
2 FROM 病例基本信息表 NATURAL JOIN 病例行程信息表
3 WHERE 省 = '河北省';

```

语句将两个表自然连接，筛选出河北省病例的信息。查询结果如图 19。

查询披露的确诊患者信息中年龄最大的患者 SQL 语句如下：

```

1 SELECT 性别，年龄，患者信息
2 FROM 病例基本信息表

```

	行程号	病例号	患者信息	日期信息	行程信息
1	1	1	确诊病例1: 女, 54岁, 冀城区增村镇南桥寨村人	2021年1月1日至10日	除1月3日12时步行到南桥寨惠兰粮油店买菜外, 1
2	2	1	确诊病例1: 女, 54岁, 冀城区增村镇南桥寨村人	1月11日	转运至冀城区指定隔离点进行集中隔离医学观察,
3	3	1	确诊病例1: 女, 54岁, 冀城区增村镇南桥寨村人	1月18日	转运至石家庄市第三医院发热门诊就诊, 核酸检测
4	4	1	确诊病例1: 女, 54岁, 冀城区增村镇南桥寨村人	1月19日	诊断为确诊病例。
5	5	10	确诊病例10: 男, 49岁, 冀城区增村镇南桥寨村。	2021年1月4日至12日	居村无外出, 期间5日、7日、9日、11日4次核酸
6	6	10	确诊病例10: 男, 49岁, 冀城区增村镇南桥寨村。	1月13日	转运至冀城区指定隔离点进行集中隔离医学观察,
7	7	10	确诊病例10: 男, 49岁, 冀城区增村镇南桥寨村。	1月19日	核酸检测呈阳性, 当日由120负压救护车转运至石
8	8	100	确诊病例16: 男, 85岁, 冀城区增村镇南桥寨村。	2020年12月30日至2021年1月10日	居村无外出, 期间1月3日、5日、7日3次核酸检

图 19: 查询结果 16

```

3 WHERE 年龄 >= ALL (
4     SELECT 年龄
5     FROM 病例基本信息表
6     WHERE 年龄 IS NOT NULL
7 )

```

语句将确诊病例的信息与其他所有病例的年龄比较, 找到大于等于所有其他患者年龄的患者。查询结果如图 20。

	性别	年龄	患者信息
1	女	95	张某某, 女, 95岁

图 20: 查询结果 17

2020 年 12 月份新增确诊患者最多的城市 SQL 语句如下:

```

1 WITH 十二月各市确诊 AS (
2     SELECT 市, COUNT(*) AS 病例数
3     FROM 病例基本信息表
4     WHERE 日期 BETWEEN '2020-12-01' AND '2020-12-31'
5     GROUP BY 市
6 )
7 SELECT *
8 FROM 十二月各市确诊
9 WHERE 病例数 >= ALL (
10     SELECT 病例数
11     FROM 十二月各市确诊
12 )

```

语句首先建立十二月各个市确诊的临时关系, 之后选出病例数大于其他所有城市的城市。查询结果如图 21。

	市	病例数
1	大连市	38

图 21: 查询结果 18

没有新增确诊病例或未披露病例信息的省份 SQL 语句如下:

```
1 SELECT 中文名称 AS 省
2 FROM 全国各省参考信息表
3 WHERE 中文名称 IS NOT NULL
4 EXCEPT
5 SELECT DISTINCT 省
6 FROM 病例基本信息表
```

用集合的差实现。查询到 21 条结果, 查询结果如图 22。

	省
1	甘肃省
2	江西省
3	浙江省
4	西藏自治区
5	河南省
6	宁夏省
7	云南省
8	江苏省

图 22: 查询结果 19

全国中高风险地区所在省中, 在 2021 年 1 月 20 日没有新增确诊信息披露的 SQL 语句如下:

```
1 SELECT DISTINCT 省
2 FROM 全国城市风险等级表
3 EXCEPT
4 SELECT DISTINCT 省
5 FROM 病例基本信息表
6 WHERE 日期 = '2021-01-20'
```

用集合的差实现。查询结果如图 23。

	省
1	吉林省
2	辽宁省

图 23: 查询结果 20

一月份国内新增患者病例最多的城市 SQL 语句如下:

```
1 WITH 一月各市确诊 AS (
2     SELECT 市, COUNT(*) AS 病例数
3     FROM 病例基本信息表
4     WHERE 日期 BETWEEN '2021-01-01' AND '2021-01-31'
5     GROUP BY 市
6 )
7 SELECT *
8 FROM 一月各市确诊
```

```

9 WHERE 病例数 >= ALL (
10     SELECT 病例数
11     FROM 一月各市确诊
12 )

```

和查询 18 一致。查询结果如图 24。

	市	病例数
1	石家庄市	788

图 24: 查询结果 21

除中美两国以外的其余国家中，进入 2021 年以来单日新增确诊病例始终不低于一万例的国家 SQL 语句如下：

```

1 WITH 排除中美的国家 AS (
2     SELECT 日期, 国家地区, SUM(累计确诊) AS 国家累计确诊
3     FROM 各国疫情数据统计表 S
4     WHERE 日期 >= '2020-12-31' AND 国家地区 NOT IN ('US', 'China') AND NOT
5         ↳ EXISTS (
6             SELECT *
7             FROM 各国疫情数据统计表 T
8             WHERE S.国家地区 = T.国家地区 AND T.省州 IS NULL
9         )
10    GROUP BY 日期, 国家地区
11 UNION
12 SELECT 日期, 国家地区, 累计确诊 AS 国家累计确诊
13 FROM 各国疫情数据统计表
14 WHERE 日期 >= '2020-12-31' AND 国家地区 NOT IN ('US', 'China') AND 省州 IS
15 ↳ NULL
16 )
17 SELECT DISTINCT 国家地区
18 FROM 排除中美的国家 AS S
19 WHERE 10000 <= ALL (
20     SELECT T2.国家累计确诊 - T1.国家累计确诊
21     FROM 排除中美的国家 AS T1 JOIN 排除中美的国家 AS T2 USING (国家地区)
22     WHERE T2.日期 = T1.日期 + 1 AND T1.国家地区 = S.国家地区
23 )

```

语句首先筛选出除了中美的国家在一月的累计确诊信息，由于有的国家的总确诊数已经给出，而另一些需要通过求各个地区确诊数的和来求出，因此要求两个集合的并。最后从这些数据中通过两个关系算出某个国家每相邻两天确诊数的差值，并且要求这些差值都大于 10000。查询结果如图 25。

9 实验总结

在实验的过程中，主要涉及的 SQL 语法错误如下：

	国家和地区
1	Russia
2	United Kingdom
3	Brazil

图 25: 查询结果 22

- 用单引号括起字符串；
- 子查询的列名和外部查询的列名重复；
- **WHERE**子句中试图访问**FROM**子句中未指明的关系；
- **WHERE**子句中试图声明别名；
- 子查询连接时未声明别名。

另外，需要观察数据足够细致才能注意到有些数据中不需要聚合求和而另一些数据中需要。

实验中，我参考 PostgreSQL 的语法说明和教材上的内容编写 SQL 语句，使我对 SQL 语法的熟悉程度大大增加，同时增强了我的英文文献阅读能力，我收获颇丰。