《数据库技术实验》报告

|  |  |  |
| --- | --- | --- |
| **第 四 次实验** | **日期：2022-10-11** | **得分：** |
| **学号：20201060351** | **姓名：张翰林** | **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. 掌握嵌套查询、集合查询和基于派生表的查询

3. 了解带有EXISTS谓词的子查询

二、实验内容

（一）实验题

1. 在Student\_Course数据库上验证课本上嵌套查询、集合查询和基于派生表查询的例子

2. 创建TPC\_H数据库并导入表Supplier、Part、PartSupp、Region、Nation、Customer、Orders、Lineitem

3. 在TPC\_H数据库的表上自行设计5个以上嵌套查询语句

4. 在TPC\_H数据库的表上自行设计5个以上集合查询语句

5. 在TPC\_H数据库的表上自行设计5个以上基于派生表的查询语句

（二）思考题

1. 如何用EXISTS/NOT EXISTS实现全称量词？

2. 如何用EXISTS/NOT EXISTS实现逻辑蕴含？

三、实验过程及实验结果

1.

(1) 嵌套查询：查询cno为2的学生：

|  |  |
| --- | --- |
| SELECT sname  from student  WHERE sno IN (          SELECT sno          from sc          WHERE cno = '2'      ); | mysql> SELECT sname  -> from student  -> WHERE sno IN (  -> SELECT sno  -> from sc  -> WHERE cno = '2'  -> );  +--------+  | sname |  +--------+  | 李四 |  | 王五 |  +--------+  2 rows in set (0.00 sec) |

(2) 嵌套查询：查找和张三同系的学生

|  |  |
| --- | --- |
| SELECT sno, sname, sdept  FROM student  WHERE sdept = (          SELECT sdept          FROM student          WHERE sname = '张三'      ); | mysql> SELECT sno, sname, sdept  -> FROM student  -> WHERE sdept = (  -> SELECT sdept  -> FROM student  -> WHERE sname = '张三'  -> );  +-------------+--------+-----------+  | sno | sname | sdept |  +-------------+--------+-----------+  | 20191060001 | 张三 | 计算机 |  | 20191060003 | 王五 | 计算机 |  | 20191060005 | 钱七 | 计算机 |  +-------------+--------+-----------+  3 rows in set (0.00 sec) |

(3) 集合查询：查找计算机系学生及年龄不大于19岁的学生

|  |  |
| --- | --- |
| -- 就是“计算机系”和“不大于19岁”两个群体的并集。  (      SELECT \*      FROM student      WHERE sdept = '计算机'  ) UNION (      SELECT \*      FROM student      WHERE sage <= 19  ); | mysql> (  -> SELECT \*  -> FROM student  -> WHERE sdept = '计算机'  -> )  -> UNION (  -> SELECT \*  -> FROM student  -> WHERE sage <= 19  -> );  +-------------+--------+------+------+--------------+  | sno | sname | ssex | sage | sdept |  +-------------+--------+------+------+--------------+  | 20191060001 | 张三 | 男 | 19 | 计算机 |  | 20191060003 | 王五 | 男 | 19 | 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) |

(4) 集合查询：查询计算机系和年龄不大于19岁学生的差集

|  |
| --- |
| -- 就是从“计算机系”学生中除去“年龄不大于19”岁的学生。MySQL中没有EXCEPT，所以只能用这种办法代替了  SELECT \* FROM student WHERE sdept = '计算机' AND sage>19; |
| mysql> SELECT \* FROM student WHERE sdept = '计算机' AND sage>19;  Empty set (0.00 sec) |

经查，student表中确实没有满足该条件的学生，所以结果为空。

(5) 基于派生表的查询：找出每个学生超过他自己选修课平均成绩的课程号

|  |  |
| --- | --- |
| SELECT sno, cno  FROM sc, (          SELECT              sno,              AVG(grade)          FROM sc          GROUP BY (sno)      ) AS Avg\_sc(avg\_sno, avg\_score)  WHERE      sc.sno = Avg\_sc.avg\_sno      AND sc.grade > Avg\_sc.avg\_score; | mysql> SELECT sno, cno  -> FROM sc, (  -> SELECT  -> sno,  -> AVG(grade)  -> FROM sc  -> GROUP BY (sno)  -> ) AS Avg\_sc(avg\_sno, avg\_score)  -> WHERE  -> sc.sno = Avg\_sc.avg\_sno  -> AND sc.grade > Avg\_sc.avg\_score;  +-------------+-----+  | sno | cno |  +-------------+-----+  | 20191060002 | 4 |  | 20191060003 | 3 |  | 20191060004 | 6 |  | 20191060005 | 5 |  +-------------+-----+  4 rows in set (0.00 sec) |

2.

使用如下命令导入：

|  |
| --- |
| mysql> create database TPC\_H;  Query OK, 1 row affected (0.01 sec)  mysql> use tpc\_h;  Database changed  mysql> source E:/Files/TIM/DataExp4.sql; |

展示导入效果：

|  |
| --- |
| mysql> show tables;  +-----------------+  | Tables\_in\_tpc\_h |  +-----------------+  | customer |  | lineitem |  | nation |  | orders |  | part |  | partsupp |  | region |  | supplier |  +-----------------+  8 rows in set (0.00 sec) |

3. 嵌套查询

(1) 查询regionkey为1的供应商

|  |  |
| --- | --- |
| SELECT name FROM supplier  WHERE nationkey IN (  SELECT nationkey  FROM nation  WHERE regionkey = 1  ); |  |

(2) 查询在lineitem中出现的order所对应的customer.name

|  |  |
| --- | --- |
| SELECT name FROM customer WHERE custkey IN (  SELECT custkey   FROM orders   WHERE orderkdy IN (  SELECT orderkey   FROM lineitem  ) ); |  |

(3) 查询在lineitem中出现的order所对应的part.name

|  |  |
| --- | --- |
| SELECT name FROM part WHERE partkey IN (  SELECT partkey   FROM lineitem ); |  |

(4) 查询在lineitem中出现的order所对应的supplier.name

|  |  |
| --- | --- |
| SELECT name FROM supplier WHERE suppkey IN (  SELECT lineitem.suppkey  FROM lineitem ); |  |

(5) 查询在lineitem中出现的order对应的nation.name

|  |  |
| --- | --- |
| SELECT name FROM nation WHERE nationkey IN (  SELECT supplier.nationkey  FROM supplier  WHERE supplier.suppkey IN (  SELECT DISTINCT suppkey   FROM lineitem  ) ); |  |

4.

(1) 查询customer中custkey<100且nationkey<100的custmer.name

(按理说应当使用(SELECT *\** FROM customer WHERE custkey < 100) INTERSECT (SELECT \* FROM customer WHERE nationkey < 100);但是经查发现MySQL没有INTERSECT，所以就使用等价的AND连接两个WHERE表示了)

|  |  |
| --- | --- |
| SELECT *\** FROM customer WHERE custkey < 100  AND nationkey < 100; |  |

(2) 查询supplier中suppkey<200或nationkey>200的supplier.name

|  |  |
| --- | --- |
| (  SELECT *\**  FROM supplier  WHERE suppkey < 200  ) UNION (  SELECT *\**  FROM supplier  WHERE nationkey > 200  ); |  |

(3) 查询part中partkey<=100和retailprice>50的差集

（同样是因为没有EXCEPT，所以使用WHERE的等价条件代替）

|  |  |
| --- | --- |
| SELECT *\** FROM part WHERE partkey <= 100  AND retailprice <= 50; |  |

(4) 查询在亚洲或大洋洲的国家的名字

|  |  |
| --- | --- |
| (  SELECT name  FROM nation  WHERE nation.regionkey = (  SELECT regionkey  FROM region  WHERE region.name = '亚洲'  ) ) UNION (  SELECT name  FROM nation  WHERE nation.regionkey = (  SELECT region.regionkey  FROM region  WHERE region.name = '大洋洲'  ) ); |  |

(5) 查询listitem中quantity<20和quantity>40的元组

|  |  |
| --- | --- |
| (SELECT *\** FROM lineitem  WHERE quantity < 20) UNION (SELECT *\** FROM lineitem  WHERE quantity > 40); |  |

5.

(1) 查询不同part的retailprice和availqty

|  |
| --- |
| SELECT availqty, retailprice FROM partsupp,  (SELECT retailprice, partkey FROM part) AS retail WHERE partsupp.partkey = retail.partkey; |
| 表格  描述已自动生成 |

(2) 查询regionkey为1的供应商

|  |
| --- |
| SELECT supplier.name, regionkey FROM supplier,  (SELECT nationkey, regionkey FROM nation) AS nation\_code WHERE supplier.nationkey = nation\_code.nationkey  AND nation\_code.regionkey = '1'; |
|  |

(3) 查询在lineitem中出现的order所对应的customer.name

|  |  |
| --- | --- |
| SELECT customer.name,  orderkdy FROM customer, (  SELECT orderkdy  FROM orders) AS od where orderkdy IN (  SELECT orderkey  FROM lineitem  ); |  |

(4) 查询supplier中suppkey<200且nationkey>200的supplier.name

|  |
| --- |
| SELECT name,suppkey,nationkey FROM (SELECT name, suppkey, nationkey FROM supplier WHERE suppkey < 200) AS part\_nation WHERE nationkey > 200; |
|  |

(5) 查询customer中custkey<100且nationkey<100的custmer.name

|  |
| --- |
| SELECT name, custkey, nationkey FROM (SELECT name, custkey, nationkey FROM customer WHERE customer.custkey < 100) AS part\_customer WHERE nationkey < 200; |
|  |

四、思考题

1.

根据，故SELECT \* FROM table WHERE P可表示为SELECT \* FROM table WHERE NOT EXISTS NOT P。

如PPT例子，“(全部课程)都选了”的学生==“不存在(存在课(没选修))”的学生==“不存在(不存在(全部课都选修了))”的学生。所以表示为 SELECT name FROM student WHERE NOT EXISTS (SELECT \* FROM Course WHERE NOT EXISTS(SELECT \* FROM sc WHERE Sno=Student.Sno AND Cno=Course.Cno));

2.

根据，以PPT例子，查询至少选修了202015122的选修课的学生，那么根据语义“202015122选了课程*y*→这些学生选了课程*y*”

则*p*：202015122选了*y*，*q*：学生*x*选了*y*，上述查询表示为：

即：查找“不存在202015122选了的课y，但x没选”的学生x，而“没选”又可以表示为NOT EXISTS。

所以表示为SELECT DISTINCT Sno FROM sc AS sc\_x WHERE NOT EXISTS(SELECT \* FROM sc AS sc\_y WHERE scy.Sno=’202015122’ AND NOT EXISTS(SELECT \* FROM sc AS sc\_z WHERE sc\_z.Sno=sc\_x.Sno AND sc\_z.Cno=sc\_y.Cno));其中最内层括号内表示202015122选的课，加了NOT EXISTS表示“不存在202015122选的课”，再往外一个NOT EXISTS表示“不存在(不存在202015122选的课)”，这样的学生便是选了202015122所有选课的同学了。