|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 数据库系统及安全  实验手册 | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | **学号：20373108** | |  |
|  |  |  |  |
|  | **姓名：张耀东** | |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **网络空间安全学院**  **2022年秋季** | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

实验二、SQL高级查询

1. 实验目的
2. 掌握多表连接、子查询和分组统计的SQL语法；
3. 熟悉SQL内、外连接的语法、使用场景和用法；
4. 了解in、exist、any、all等谓词的使用方法；
5. 了解触发器的基本原理和编写方法。
6. 实验要求
7. 作业模板可基于本实验手册，在实验内容后面直接填写实验报告。
8. 作业提交方式：电子版(rar/zip打包)，文件命名格式：[学号]-[姓名]-实验[序号1位数，1--9].rar/zip；作业文件建议使用word或者pdf格式，不接受拍照图片版本。
9. 团队项目：[项目名称]-[小组序号2位数, 01--99].rar/zip, 压缩文件内须包含一个组员学号、姓名和具体分工说明的txt文件[文件名：小组组成与任务分工.txt]。
10. 作业提交到北航网盘：共享目录“DBMS-2022-作业”上传

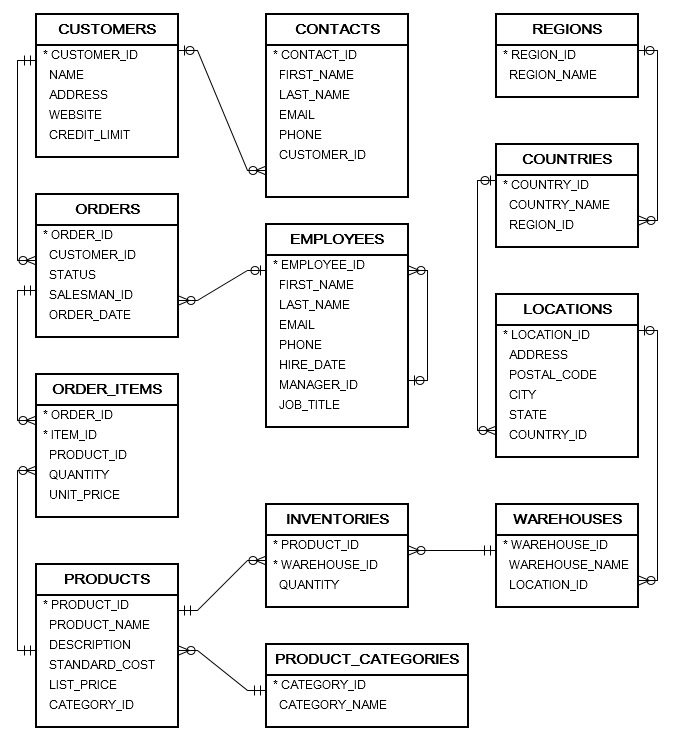
https://bhpan.buaa.edu.cn:443/link/53C3AF672185198F3381B8E0F83229DF 密码：z0sD

1. 每一个实验内容，根据要求，进行实际操作，并把具体的步骤记录下来，如给出数据查询/修改等的SQL语句，查询结果截图后附在后面（如果数据很多，可只截第一页页面，不需要把全部的查询结果截图）；安装配置类的，可以只截重要的配置页面，以及测试成功的页面。

或者可以参考本实验提供的sqlplus的spool.sql文件，将结果输出到外部的一个html文件，随作业一块提交。

1. 实验作业期限：自本实验日期起，3周之内，下一次实验课之前。
2. 实验内容

根据oracle 的sample db, 进行如下的SQL练习：



**图1 测试数据库ERD**

1. 查询所有仓库的仓库名称、地址、邮政编码、城市、州名、国家名称和区域名称。

|  |
| --- |
| **SELECT** **DISTINCT** warehouses.WAREHOUSE\_NAME,locations.ADDRESS,locations.postal\_code,locations.city,locations.state,countries.country\_name,regions.region\_name  **FROM** warehouses **NATURAL** **JOIN** locations **NATURAL** **JOIN** countries **NATURAL** **JOIN** regions; |
|  |

1. 查询所有客户联系人（contacts）和雇员的信息，按照以下格式显示：

姓名（first\_name空格last\_name）、email、人员类别（客户联系人显示“contact”，雇员显示“employee”），并按照姓名升序排序；

查询所有客户联系人（contacts）和雇员的姓氏信息，如果有重复的话，如何去重。

|  |
| --- |
| **SELECT** **CONCAT**(first\_name,' ',LAST\_name) 姓名,email,"employee" 类别  **FROM** employees  **UNION**  **SELECT** **CONCAT**(first\_name,' ',LAST\_name) 姓名,email,"contact" 类别  **FROM** contacts  **ORDER** **BY** 姓名; |
|  |
| **SELECT** **DISTINCT** 姓  **FROM** (  **SELECT** LAST\_name 姓  **FROM** employees  **UNION**  **SELECT** LAST\_name 姓  **FROM** contacts  )**AS** peoples  **ORDER** **BY** 姓; |
|  |

1. 查询有销售人员参与成交的订单客户信息，结果显示客户联系人姓名、客户公司名称和销售代表姓名。

|  |
| --- |
| **SELECT** **DISTINCT** **CONCAT**(contacts.first\_name,' ',contacts.last\_name) 客户联系人姓名,customers.**name** 客户公司名称,**CONCAT**(employees.first\_name,' ',employees.last\_name) 销售代表名称  **FROM** contacts,customers,orders,employees  **WHERE**  contacts.customer\_id = customers.customer\_id **AND** customers.customer\_id=orders.customer\_id **and** orders.salesman\_id = employees.employee\_id  **ORDER** **BY** **CONCAT**(contacts.first\_name,' ',contacts.last\_name);  *-- AND orders.`status`="Shipped" 应该加上这一项，但是答案却没有加* |
|  |

1. 查询2017年销售额高于100万的销售员，显示ID和姓名。

|  |
| --- |
| **SELECT** employees.employee\_id,**CONCAT**(employees.first\_name,employees.last\_name),**SUM**(salesman\_price.price) sum\_price  **FROM** (  **SELECT** orders.salesman\_id salesman\_id,order\_items.quantity \* order\_items.unit\_price price  **FROM** orders,order\_items  **WHERE** orders.`STATUS` <> 'Canceled' **AND** **year**(orders.ORDER\_DATE) = '2017' **AND** orders.ORDER\_ID = order\_items.order\_id  ) **as** salesman\_price,employees  **WHERE** salesman\_price.salesman\_id = employees.employee\_id  **GROUP** **BY** employees.employee\_id  **HAVING** sum\_price>1000000.0  **ORDER** **BY** employees.employee\_id; |
|  |

1. 查询没有任何人购买的商品名称和商品描述。

|  |
| --- |
| **SELECT** products.product\_name 商品名称,products.**description**  **FROM** products  **WHERE** products.product\_id **NOT** **IN** (  **SELECT** **DISTINCT** order\_items.product\_id  **FROM** orders , order\_items  **WHERE** orders.order\_id = order\_items.order\_id  ); |
|  |

1. 查询45号客户46号客户都订购过的产品信息（产品ID和名称）(提示：可以采用集合运算或者outer join)。

|  |
| --- |
| **SELECT** products1.product\_id id,products1.product\_name **name**  **FROM** orders orders1,orders orders2,order\_items order\_items1,order\_items order\_items2,products products1,products products2  **WHERE**  orders1.customer\_id=45 **AND**  orders2.customer\_id=46 **AND**  orders1.order\_id=order\_items1.order\_id **AND**  orders2.order\_id=order\_items2.order\_id **AND**  order\_items1.product\_id=products1.product\_id **AND**  order\_items2.product\_id=products2.product\_id **AND**  products1.product\_id=products2.product\_id; |
|  |

1. 查询所有客户订单状态为”Shipped”的如下统计信息：每一位客户的ID、姓名、订单数量、订单总金额和订单平均金额。

|  |
| --- |
| **SELECT** **DISTINCT** customers.customer\_id 客户ID, customers.**name** 客户姓名, **COUNT**(A.order\_id),**SUM**(A.sum\_p),**AVG**(A.sum\_p)  **FROM** customers **JOIN** (  **SELECT** **DISTINCT** orders.customer\_id ID, orders.order\_id,**SUM**(order\_items.quantity\*order\_items.unit\_price) sum\_p  **FROM** orders **JOIN** order\_items **ON** orders.order\_id = order\_items.order\_id  **WHERE** orders.`status`='Shipped'  **GROUP** **BY** orders.order\_id  ) **AS** A **ON** customers.customer\_id = A.ID  **GROUP** **BY** customers.customer\_id  **ORDER** **BY** customers.customer\_id; |
|  |

1. 查询所有客户的每个年度的商品采购总金额，按照以下格式显示和排序：

客户ID、客户名称、年度、采购金额（降序）。

|  |
| --- |
| **SELECT** customers.customer\_id 客户ID,customers.**name** 客户名称,customer\_price.year\_time 年度,customer\_price.total\_price 采购金额  **FROM** (  **SELECT** **SUM**(order\_items.quantity\*order\_items.unit\_price) total\_price,**YEAR**(orders.order\_date) year\_time,orders.customer\_id id  **FROM** orders,order\_items  **WHERE** orders.order\_id = order\_items.order\_id **AND** orders.`status`<>"Canceled"  **GROUP** **BY** **YEAR**(orders.order\_date),orders.customer\_id  ) **AS** customer\_price,customers  **WHERE** customer\_price.id = customers.customer\_id  **GROUP** **BY** customers.customer\_id,customer\_price.year\_time  **ORDER** **BY** 采购金额 **DESC；** |
|  |

1. 查询没有库存的商品id和商品名称。

|  |
| --- |
| **SELECT** **DISTINCT** products.product\_id 商品id,products.product\_name 商品名称  **FROM** products  **WHERE** products.product\_id **NOT** **IN**(  **SELECT** inventories.product\_id  **FROM** inventories  ); |
|  |

1. 查询没有任何订单的销售人员名单(id, first name, last name)，要求分别使用IN 、EXISTS谓词 和外连接。

|  |
| --- |
| **SELECT** **DISTINCT** employees.employee\_id,employees.first\_name,employees.last\_name  **FROM** employees  **WHERE** employees.employee\_id **NOT** **IN**  (  **SELECT** **DISTINCT** employees.employee\_id  **FROM** employees,orders  **WHERE** employees.employee\_id = orders.salesman\_id  ); |
| **SELECT** **DISTINCT** employees0.employee\_id,employees0.first\_name,employees0.last\_name  **FROM** employees employees0  **WHERE** **NOT** **EXISTS**(  **SELECT** \*  **FROM** orders  **WHERE** orders.salesman\_id = employees0.employee\_id  ); |
| **SELECT** **DISTINCT** employees.employee\_id,employees.first\_name,employees.last\_name  **FROM** employees **LEFT** **JOIN** orders **ON** employees.employee\_id=orders.salesman\_id  **WHERE** orders.order\_id **IS** **NULL**; |
|  |

1. 统计每个仓库按产品类型（Category）的库存商品数量，显示仓库名称、产品类型和库存数量，并按照仓库名称和产品类型对结果进行排序。

|  |
| --- |
| **SELECT** warehouses.warehouse\_name 仓库名称,product\_categories.category\_name 产品类型,**SUM**(inventories.quantity) 库存数量  **FROM** warehouses,product\_categories,inventories,products  **WHERE** warehouses.warehouse\_id=inventories.warehouse\_id **AND** inventories.product\_id=products.product\_id **AND** products.category\_id=product\_categories.category\_id  **GROUP** **BY** 仓库名称,产品类型  **ORDER** **BY** 仓库名称,产品类型; |
|  |

1. 查询采购了至少与100号订单的产品相同的客户信息（ID与客户名称）。

|  |
| --- |
| 若理解为某位客户的所有订单中包含100号订单的话，如下  **SELECT** **DISTINCT** customers0.customer\_id ID,customers0.**name** 客户名称  **FROM** (  **SELECT** order\_items.product\_id id  **FROM** order\_items,orders  **WHERE** order\_items.order\_id = 100  ) **AS** pid,customers customers0  **WHERE** **NOT** **EXISTS**(  **SELECT** \*  *-- FROM order\_items*  **WHERE** pid.id **NOT** **IN**(  **SELECT** order\_items.product\_id  **FROM** orders,order\_items  **WHERE** customers0.customer\_id = orders.customer\_id **AND** order\_items.product\_id  )  ); |
|  |
| 若理解为某位客户的某次订单中包含100号订单的话，如下  **SELECT** **DISTINCT** customers0.customer\_id ID,customers0.**name** 客户名称  **FROM** customers customers0,orders orders0  **WHERE** customers0.customer\_id = orders0.customer\_id **AND** orders0.`status`='Shipped' **AND** **NOT** **EXISTS**  (  **SELECT** \*  **FROM** order\_items order\_items100  **WHERE** order\_items100.order\_id = 100 **AND** order\_items100.product\_id **NOT** **IN**  (  **SELECT** order\_items0.product\_id  **FROM** order\_items order\_items0  **WHERE** orders0.order\_id = order\_items0.order\_id  )  ); |
|  |

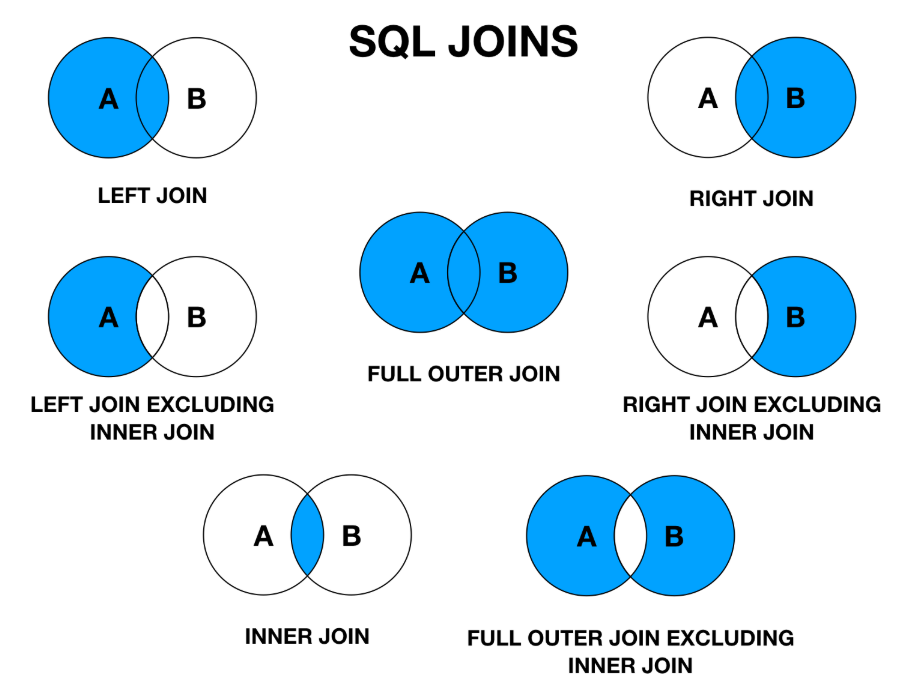
1. 查询管理员工数量最多的经理信息。

|  |
| --- |
| **SELECT** m.id 经理ID,**CONCAT**(employees.first\_name,' ',employees.last\_name) 经理姓名,employees.email 经理邮件,employees.phone 经理电话,**DATE\_FORMAT**(employees.hire\_date,'%Y-%M-%D') 雇佣日期,employees.job\_title 职位  **FROM** (  **SELECT** manager.employee\_id id,**COUNT**(\*) amount  **FROM** employees employee,employees manager  **WHERE** employee.manager\_id = manager.employee\_id  **GROUP** **BY** manager.employee\_id  ) **AS** m,employees  **WHERE** m.id = employees.employee\_id **AND** m.amount >= **ALL**  (  **SELECT** **COUNT**(\*) amount  **FROM** employees employee,employees manager  **WHERE** employee.manager\_id = manager.employee\_id  **GROUP** **BY** manager.employee\_id  ); |
|  |

1. 使用SQL将COUNTRIES的数据复制到COUNTRY\_A，然后将region=4的国家从COUNTRY\_A中删除；使用SQL将COUNTRIES的数据复制到COUNTRY\_B，然后将region=2的国家从COUNTRY\_B中删除。

根据下图，基于上述修改后的COUNTRY数据，假设A=COUNTRY\_A，B=COUNTRY\_B，给出对应的SQL查询结果，并解释结果意义。

|  |
| --- |
| **CREATE** **TABLE** country\_a **SELECT** \* **FROM** countries;  **CREATE** **TABLE** country\_b **SELECT** \* **FROM** countries;  **DELETE** **FROM** country\_a **WHERE** country\_a.region\_id = 2;  **DELETE** **FROM** country\_b **WHERE** country\_b.region\_id = 4; |
| LEFT JOIN如下  **SELECT** \*  **FROM** country\_a **LEFT** **JOIN** country\_b **ON** country\_a.country\_id = country\_b.country\_id;  解释：选择所有包含在A中的数据。 |
|  |
| LEFT JOIN EXCLUDING INNER JOIN如下  **SELECT** \*  **FROM** country\_a **LEFT** **JOIN** country\_b **ON** country\_a.country\_id = country\_b.country\_id  **WHERE** country\_b.country\_id **IS** **NULL**;  解释：选择所有包含在A中但不包含在B中的数据。 |
|  |
| INNER JOIN如下  **SELECT** \*  **FROM** country\_a **INNER** **JOIN** country\_b **ON** country\_a.country\_id = country\_b.country\_id;  解释：选择所有既包含在A中也包含在B中的数据。 |
|  |
| RIGHT JOIN如下  **SELECT** \*  **FROM** country\_a **RIGHT** **JOIN** country\_b **ON** country\_a.country\_id = country\_b.country\_id;  解释：选择所有包含在B中的数据。 |
|  |
| RIGHT JOIN EXCLUDING INNER JOIN如下  **SELECT** \*  **FROM** country\_a **RIGHT** **JOIN** country\_b **ON** country\_a.country\_id = country\_b.country\_id  **WHERE** country\_a.country\_id **IS** **NULL**;  解释：选择所有包含在B中但不包含在A中的数据。 |
|  |
| FULL OUTER JOIN EXCLUDING INNER JOIN如下  **SELECT** \*  **FROM** country\_a **LEFT** **JOIN** country\_b **ON** country\_a.country\_id = country\_b.country\_id  **UNION**  **SELECT** \*  **FROM** country\_a **RIGHT** **JOIN** country\_b **ON** country\_a.country\_id = country\_b.country\_id;  解释：选择所有要么包含在A中，要么包含在B中的数据。 |
|  |
| FULL OUTER JOIN如下  **SELECT** \*  **FROM** country\_a **LEFT** **JOIN** country\_b **ON** country\_a.country\_id = country\_b.country\_id  **WHERE** country\_b.country\_id **IS** **NULL**  **UNION**  **SELECT** \*  **FROM** country\_a **RIGHT** **JOIN** country\_b **ON** country\_a.country\_id = country\_b.country\_id  **WHERE** country\_a.country\_id **IS** **NULL**;  解释：选择所有包含在A中或包含在B中的数据。 |
|  |



**图2 SQL连接类型**

1. 将orders表复制一份结构和数据到order\_trg，在新建的order\_trg表上创建一个after update触发器，当把STATUS为“Pending”值的记录修改为“Shipped”时，要求在触发器中将每一条订单记录的修改结果记录到一个日志表update\_log，该表需要保存订单ID、修改前的STATUS值、修改后的STATUS值、当前的用户名和修改时间（精确到秒）。

|  |
| --- |
| **CREATE** **TABLE** orders\_trg **SELECT** \* **FROM** orders;  **CREATE** **TABLE** update\_log(  order\_id **BIGINT**,  old\_status **VARCHAR**(20),  new\_status **VARCHAR**(20),  log\_user **VARCHAR**(50),  log\_time **DATETIME**,  **PRIMARY** **KEY**(order\_id,log\_user,log\_time)  );  delimiter $  **CREATE** **TRIGGER** or\_t  **AFTER** **UPDATE** **ON** orders\_trg **FOR** **EACH** **ROW**  **BEGIN**  **if**(**OLD**.`status`='Pending' **AND** **NEW**.`status`='Shipped') **then**  **INSERT** **INTO** update\_log **value**(**OLD**.order\_id,**OLD**.`status`,**NEW**.`status`,**USER**(),**NOW**());  **END** **if**;  **END**; |
| **UPDATE** orders\_trg **SET** orders\_trg.`status` = 'Shipped' **WHERE** orders\_trg.order\_id=1; |
|  |

1. 实验总结与建议

答：本次实验中出现多次与题目意思理解误差的地方，如“查询有销售人员参与成交的订单客户信息”中，订单被“Canceled”后应当是未成交，却不用排除“Canceled”的订单，在自己的结果与参考输出中纠结了许久，以找到与参考相同或相近的解释以及sql语句。本次实验也加深了我对sql语句的使用，帮助我学习使用创建、查询、插入，更新表格以及创建使用触发器。