**实验训练2：数据查询操作**

**实验目的：**

基于实验1创建的汽车用品网上商城数据库Shopping，理解MySQL运算符、函数、谓词，练习Select语句的操作方法。

**实验内容：**

1.单表查询

【实验2.1】字段查询

（1）查询商品名称为“挡风玻璃”的商品信息。

分析：商品信息存在于商品表，而且商品表中包含商品名称此被查询信息，因此这是只需要涉及一个表就可以完成简单单表查询。

（2）查询ID为1的订单。

分析：所有的订单信息存在于订单表中，而且订单用户ID也存在于此表中，因此这是只需要查询订单表就可以完成的查询。

【实验2.2】多条件查询

查询所有促销的价格小于1000的商品信息。

分析：此查询过程包含两个条件，第一个是是否促销，第二个是价格，在商品表中均有此信息，因此这是一个多重条件的查询。

【实验2.3】DISTINCT

（1）查询所有对商品ID为1的商品发表过评论的用户ID。

分析：条件和查询对象存在于评论表中，对此商品发表过评论的用户不止一个，而且一个用户可以对此商品发表多个评论，因此，结果需要进行去重，这里使用DISTINCT实现。

（2）查询此汽车用品网上商城会员的创建时间段，1年为一段。

分析：通过用户表可以完成查询，每年可能包含多个会员，如果把此表中的创建年份都列出来会有重复，因此使用DISTINCT去重。

【实验2.4】ORDER BY

（1）查询类别ID为1的所有商品，结果按照商品ID降序排列。

分析：从商品表中可以查询出所有类别ID为1的商品信息，结果按照商品ID的降序排列，因此使用ORDER BY语句，降序使用DESC关键字。

（2）查询今年新增的所有会员，结果按照用户名字排序。

分析：在用户表中可以完成查询，创建日期条件设置为今年，此处使用语句ORDER BY。

【实验2.5】GROUP BY

（1）查询每个用户的消费总金额（所有订单）。

分析：订单表中包含每个订单的订单总价和用户ID。现在需要将每个用户的所有订单提取出来分为一类，通过SUM()函数取得总金额。此处使用GROUP BY语句和SUM()函数。

（2）查询类别价格一样的各种商品数量总和。

分析：此查询中需要对商品进行分类，分类依据是同类别和价格，这是“多列分组”，较上一个例子更为复杂。

2.聚合函数查询

【实验2.6】COUNT()

（1）查询类别的数量。

分析：此查询利用COUNT()函数，返回指定列中值的数目，此处指定列是类别表中的ID（或者名称均可）。

（2）查询汽车用品网上商城的每天的接单数。

分析：订单相关，此处使用聚合函数COUNT()和Group by 子句。

【实验2.7】 SUM()

查询该商城每天的销售额。

分析：在订单表中，有一列是订单总价，将所有订单的订单总价求和，按照下单日期分组，使用SUM()函数和Group by子句。

【实验2.8】AVG()

（1）查询所有订单的平均销售金额。

分析：同上一个相同，还是在订单表中，依然取用订单总价列，使用AVG()函数，对指定列的值求平均数。

【实验2.9】MAX()

（1）查询所有商品中的数量最大者。

分析：商品的数量信息存在于商品表中，此处查询应该去商品表，在商品数量指定列中求值最大者。使用MAX()函数。

（2）查询所有用户按字母排序中名字最靠前者。

分析：MAX()或者MIN()也可以用在文本列，以获得按字母顺序排列的最高或者最低者。同上一个实验一样，使用MAX()函数。

【实验2.10】MIN()

（1）查询所有商品中价格最低者。

分析：同MAX（）用法相同，找到表和列，使用MIN()函数。

3.连接查询

【实验2.11】内连接查询

（1）查询所有订单的发出者名字。

分析：此处订单的信息需要从订单表中得到，订单表中主键是订单号，外键是用户ID，同时查询需要得到订单发出者的姓名，也就是用户名，因此需要将订单表和用户表通过用户ID进行连接。使用内连接的(INNER) JOIN语句。

（2）查询每个用户购物车中的商品名称。

分析：购物车中的信息可以从购物车表中得到，购物车表中有用户ID和商品ID两项，通过这两项可以与商品表连接，从而可以获得商品名称。与上一个实验相似，此查询使用(INNER) JOIN语句。

【实验2.12】外连接查询

（1）查询列出所有用户ID，以及他们的评论，如果有的话。

分析：此查询首先需列出所有用户ID，如果参与过评论的话，再列出相关的评论。此处使用外查询中的LEFT (OUTER) JOIN语句，注意需将全部显示的列名写在JOIN语句左边。

（2）查询列出所有用户ID，以及他们的评论，如果有的话。

分析：依然是上一个实验，还可以使用RIGHT (OUTER) JOIN语句，注意需将全部显示的列名写在JOIN语句右边。

【实验2.13】复合条件连接查询

（1）查询用户ID为1的客户的订单信息和客户名。

分析：复合条件连接查询是在连接查询的过程中，通过添加过滤条件，限制查询的结果，使查询的结果更加准确。此查询需在内查询的基础上加上另一个条件，用户iD为1，使用AND语句添加精确条件。

（2）查询每个用户的购物车中的商品价格，并且按照价格顺序排列。

分析：此查询需要先使用内连接对商品表和购物车表进行连接，得到商品的价格，在使用ORDER BY语句对价格进行顺序排列。

4.嵌套查询

【实验2.14】 IN

（1）查询订购商品ID为1的订单ID，并根据订单ID查询发出此订单的用户ID。

分析：此查询需要使用IN关键字进行子查询，子查询是通过SELECT语句在订单明细表中先确定此订单ID，在通过SELECT在订单表中查询到用户ID。

（2）查询订购商品ID为1的订单ID，并根据订单ID查询未发出此订单的用户ID。

分析：此查询和前一个实验相似，只是需使用NOT IN语句。

【实验2.15】比较运算符

（1）查询今年新增会员的订单，并且列出所有订单总价小于100的订单ID。

分析：此查询需要使用嵌套，子查询需先查询用户表得到今年创建的用户信息，在将用户ID匹配找打订单信息，其中使用比较运算符提供订单总价小于100的条件。

（2）查询所有订单商品数量总和小于100的商品ID，并将不在此商品所在类别的其他类别的ID列出来。

分析：此查询需要进行嵌套查询，子查询过程需要使用到SUM()函数和GROUP BY求出同种商品的所有被订数量，使用比较运算符得到数量总和小于100的商品ID，再使用比较运算符“不等于”得到非此商品所在类的类别ID。

【实验2.16】EXISTS

（1）查询表中是否存在用户ID为100的用户，如果存在，列出此用户的信息。

分析：EXISTS关键字后面的参数是一个任意的子查询，系统对于查询进行运算以判断它是否返回行，如果至少返回一行，那以EXISTS的结果为TRUE，此时外层查询语句将进行查询。此查询需要对用户ID进行EXIST操作。

（2）查询表中是否存在类别ID为100的商品类别，如果存在，列出此类别中商品价格小于5的商品ID。

分析：与上一个实验相似，此实验在外查询过程添加了比较运算符。

【实验2.17】ANY

查询所有商品表中价格比订单表中商品ID对应的价格大的商品ID。

分析：ANY关键字在一个比较操作符的后面，表示若与子查询返回的任何值比较为TRUE，则返回TRUE。此处使用ANY来引出内查询。

【实验2.18】ALL

查询所有商品表中价格比订单表中所有商品ID对应的价格大的商品ID。

分析：使用ALL时需要同时满足所有内层查询的条件。ALL关键字在一个比较操作符的后面，表示与子查询返回的所有值比较为TRUE，则返回TRUE。此处使用ALL来引出内查询。

【实验2.19】集合查询

（1）查询所有价格小于5的商品，查询类别ID为1和2的所有商品，使用UNION连接查询结果。

分析：由前所述，UNION将多个SELECT语句的结果组合成一个结果集合，第1条SELECT语句查询价格小于5的商品，第2条SELECT语句查询类别ID为1和2的商品，使用UNION将两条SELECT语句分隔开，执行完毕之后把输出结果组合为单个的结果集，并删除重复的记录。

（2）查询所有价格小于5的商品，查询类别ID为1和2的所有商品，使用UNION ALL连接查询结果。

分析：使用UNION ALL包含重复的行，在前面的例子中，分开查询时，两个返回结果中有相同的记录，使用UNION会自动去除重复行。UNION ALL从查询结果集中自动要返回所有匹配行，而不进行删除。

**实验要求：**

1．所有操作必须通过MySQL Workbench完成；

2．每执行一种查询语句后，均要求通过MySQL Workbench查看执行结果；

3．将操作过程以屏幕抓图的方式拷贝，形成实验文档。