**数据库课程项目测试用例**

**系统管理模块**

1. CREATE DATABASE *orderDB*;

创建名为orderDB的数据库。

1. DROP DATABASE *orderDB*;

删除名为orderDB的数据库。

1. USE *orderDB*;

当前数据库切换为orderDB。

1. SHOW TABLES;

列出当前数据库包含的所有表。

1. CREATE TABLE *customer*(

*id int(10) NOT NULL,*

*name varchar(25) NOT NULL,*

*gender varchar(1) NOT NULL,*

*PRIMARY KEY(id)*

);

创建名为customer的表，它包含三个字段id、name和gender，其中id是主键。这三个字段的数据类型分别为整型、字符串和字符串，并且都不允许为空。

1. DROP TABLE *customer*;

删除名为customer的表。

1. DESC *customer*;

列出customer表的模式(schema)信息。

**查询解析模块**

1. INSERT INTO *customer* VALUES *(300001, ‘CHAD CABELLO’, ‘F’)*;

在customer表中插入一条数据(300001, CHAD CABELLO, ‘F’)。如果主键出现重复，应报错，阻止插入数据。

1. INSERT INTO *orders* VALUES *(315000,200001,’eight’)*;

orders表的quantity字段的数据类型为整型，出现该语句应报错，提示quantity字段数据类型必须是整型，阻止插入数据。

1. DELETE FROM *publisher* WHERE *state=’CA’*;

删除所有加州的出版商。

1. UPDATE *book* SET *title=’Nine Times Nine’* WHERE *authors=’Anthony Boucher’*;

把作者Anthony Boucher的书的书名改为Nine Times Nine。

1. SELECT *\** FROM *publisher* WHERE *nation=’CA’*;

列出所有加州出版商的信息。

1. SELECT *title* FROM *book* WHERE *authors is null*;

列出authors字段为空的记录的书名。

1. SELECT *book.title,orders.quantity* FROM *book,orders* WHERE *book.id=orders.book\_id* AND orders.quantity>8;

列出订购数大于8本的书的书名以及相应的订购数量。

附加模块：

1. 索引模块

CREATE INDEX *customer(name)*;

为customer表的name字段创建索引。

DROP INDEX *customer(name)*;

删除customer表的name字段的索引。

2. 域完整性约束，建表时的check关键字。

CREATE TABLE customer (

id int(10) NOT NULL,

name varchar(25) NOT NULL,

*gender varchar(1) NOT NULL,*

CHECK (gender in(‘F’,’M’)),

PRIMARY KEY (id)

);

3. 外键约束。

UPDATE *book* SET *publisher\_id=’106001’* WHERE *title=’ Standing in the Shadows’*

把title字段为Standing in the Shadows的这条记录的publisher\_id字段更新为106001。如果publisher表中没有id为106001的记录，应报错，提示违反外键约束，阻止更新数据。

1. 模糊匹配。

SELECT *title,authors* FROM *book* WHERE *authors like ’%David%’*;

列出作者是David的书的书名以及作者的全称。

1. 三个表以上的连接。

SELECT *customer.name,book.title,orders.quantity* FROM *customer,book,orders* WHERE *orders.customer\_id=customer.id AND orders.book\_id=book.id AND orders.quantity > 9*;

列出订购数量大于9的订单中顾客的姓名、书名以及相应的订购数量。

1. 聚集查询。

SELECT SUM(quantity) FROM orders; 求出所有订单的订购总量。

SELECT AVG(quantity) FROM orders; 求出每个订单的平均订购量。

SELECT MAX(quantity) FROM orders; 求出所有订单中最大订购量。

SELECT MIN(quantity) FROM orders; 求出所有订单中最小订购量。

1. 分组聚集查询。

SELECT customer\_id,SUM(quantity) FROM orders GROUP BY customer\_id;

求出所有订单中每个顾客订购书的总量。