《数据库系统原理》实验报告（2）					
题目：DML 语言实验					
学号	2352018	姓名	刘彦	日期	2025.04.16

实验环境：

Docker-desktop 4.40.0

oceanbase-ce 4.3.5.1

实验步骤及结果截图：

⑴创建实验所需的数据库并使用

create database exp2;

use exp2;

```
sh-4.4# obclient -uroot@sys -h127.1 -P2881;
Welcome to the OceanBase.  Commands end with ; or \g.
Your OceanBase connection id is 3221498432
Server version: OceanBase_CE 4.3.5.1 (r101000042025031818-b6d5706eb3d2c5f501c7fa646ddbf32f3dc87069) (Built Mar 18 2025 18:13:36)

Copyright (c) 2000, 2018, OceanBase and/or its affiliates. All rights reserved.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

obclient(root@sys)[(none)]> create database exp2;
Query OK, 1 row affected (0.042 sec)

obclient(root@sys)[(none)]> use exp2;
Database changed
obclient(root@sys)[exp2]>
```

⑵参照表格内容，在数据库中建立图书借阅管理系统相关数据表

①表 category

```
CREATE TABLE category (
    No INTEGER NOT NULL,
    Name VARCHAR(50) NOT NULL,
    PRIMARY KEY (No)
);
```

```
obclient(root@sys)[exp2]> CREATE TABLE category (
    ->     No INTEGER NOT NULL,
    ->     Name VARCHAR(50) NOT NULL,
    ->     PRIMARY KEY (No)
    -> );
Query OK, 0 rows affected (0.049 sec)

obclient(root@sys)[exp2]> desc category;
+-------+-------------+------+------+---------+-------+
| Field | Type        | Null | Key  | Default | Extra |
+-------+-------------+------+------+---------+-------+
| No    | int(11)     | NO   | PRI  | NULL    |       |
| Name  | varchar(50) | NO   |      | NULL    |       |
+-------+-------------+------+------+---------+-------+
2 rows in set (0.012 sec)
```

②表 books

```
CREATE TABLE books (
    No INTEGER NOT NULL,
    Title VARCHAR(100) NOT NULL,
    Author VARCHAR(50) NOT NULL,
    publication_year INTEGER NOT NULL,
    C_no INTEGER NOT NULL,
    PRIMARY KEY (No),
```

```
    FOREIGN KEY (C_no) REFERENCES category(No)
);
```

```
obclient(root@sys)[exp2]> CREATE TABLE books (
    ->     No INTEGER NOT NULL,
    ->     Title VARCHAR(100) NOT NULL,
    ->     Author VARCHAR(50) NOT NULL,
    ->     publication_year INTEGER NOT NULL,
    ->     C_no INTEGER NOT NULL,
    ->     PRIMARY KEY (No),
    ->     FOREIGN KEY (C_no) REFERENCES category(No)
    -> );
Query OK, 0 rows affected (0.081 sec)

obclient(root@sys)[exp2]> desc books;
+------------------+-------------+------+------+---------+-------+
| Field            | Type        | Null | Key  | Default | Extra |
+------------------+-------------+------+------+---------+-------+
| No               | int(11)     | NO   | PRI  | NULL    |       |
| Title            | varchar(100) | NO   |      | NULL    |       |
| Author           | varchar(50)  | NO   |      | NULL    |       |
| publication_year | int(11)     | NO   |      | NULL    |       |
| C_no             | int(11)     | NO   |      | NULL    |       |
+------------------+-------------+------+------+---------+-------+
5 rows in set (0.011 sec)
```

③表 readers

```
CREATE TABLE readers (
    No INTEGER NOT NULL,
    Name VARCHAR(50) NOT NULL,
    Gender VARCHAR(10) NOT NULL CHECK (Gender IN ('Male', 'Female')),
    Tel VARCHAR(15) NOT NULL,
    PRIMARY KEY (No)
);
```

```
obclient(root@sys)[exp2]> CREATE TABLE readers (
    ->     No INTEGER NOT NULL,
    ->     Name VARCHAR(50) NOT NULL,
    ->     Gender VARCHAR(10) NOT NULL CHECK (Gender IN ('Male', 'Female')),
    ->     Tel VARCHAR(15) NOT NULL,
    ->     PRIMARY KEY (No)
    -> );
Query OK, 0 rows affected (0.062 sec)

obclient(root@sys)[exp2]> desc readers;
+--------+-------------+------+------+---------+-------+
| Field  | Type        | Null | Key  | Default | Extra |
+--------+-------------+------+------+---------+-------+
| No     | int(11)     | NO   | PRI  | NULL    |       |
| Name   | varchar(50) | NO   |      | NULL    |       |
| Gender | varchar(10) | NO   |      | NULL    |       |
| Tel    | varchar(15) | NO   |      | NULL    |       |
+--------+-------------+------+------+---------+-------+
4 rows in set (0.010 sec)
```

④表 borrow_records

```
CREATE TABLE borrow_records (
    No INTEGER NOT NULL,
    B_no INTEGER NOT NULL,
    R_no INTEGER NOT NULL,
    Borrow_date DATE NOT NULL,
    Return_date DATE,
    PRIMARY KEY (No),
    FOREIGN KEY (B_no) REFERENCES books(No),
    FOREIGN KEY (R_no) REFERENCES readers(No)
```

);

```
obclient(root@sys)[exp2]> CREATE TABLE borrow_records (
    ->     No INTEGER NOT NULL,
    ->     B_no INTEGER NOT NULL,
    ->     R_no INTEGER NOT NULL,
    ->     Borrow_date DATE NOT NULL,
    ->     Return_date DATE,
    ->     PRIMARY KEY (No),
    ->     FOREIGN KEY (B_no) REFERENCES books(No),
    ->     FOREIGN KEY (R_no) REFERENCES readers(No)
    -> );
Query OK, 0 rows affected (0.108 sec)

obclient(root@sys)[exp2]> desc borrow_records;
+-------------+---------+------+------+---------+-------+
| Field       | Type    | Null | Key  | Default | Extra |
+-------------+---------+------+------+---------+-------+
| No          | int(11) | NO   | PRI  | NULL    |       |
| B_no        | int(11) | NO   |      | NULL    |       |
| R_no        | int(11) | NO   |      | NULL    |       |
| Borrow_date | date    | NO   |      | NULL    |       |
| Return_date | date    | YES  |      | NULL    |       |
+-------------+---------+------+------+---------+-------+
5 rows in set (0.010 sec)
```

⑶在数据表中插入内容

INSERT INTO category (No, Name) VALUES
(1, '计算机'),
(2, '文学'),
(3, '历史'),
(4, '科学');
INSERT INTO books (No, Title, Author, publication_year, C_no) VALUES
(1, 'Python 编程实战', '张三', 2021, 1),
(2, '百年孤独', '加西亚·马尔克斯', 1967, 2),
(3, '明朝那些事儿', '当年明月', 2006, 3),
(4, '时间简史', '霍金', 1988, 4),
(5, 'Java 核心技术', 'K.S.霍斯特曼', 2019, 1);
INSERT INTO readers (No, Name, Gender, Tel) VALUES
(1, '小明', 'Male', '13800138000'),
(2, '小红', 'Female', '13900139000'),
(3, '小刚', 'Male', '13600136000'),
(4, '小美', 'Female', '13700137000'),
(5, '小强', 'Male', '13300133000');
INSERT INTO borrow_records (No, B_no, R_no, Borrow_date, Return_date) VALUES
(1, 1, 1, '2024-10-01', '2024-10-20'),
(2, 2, 2, '2024-10-05', '2024-10-25'),
(3, 1, 3, '2024-10-10', '2024-10-30'),
(4, 3, 4, '2024-10-15', '2024-11-01'),
(5, 2, 5, '2024-10-20', NULL);

```
obclient(root@sys)[exp2]> INSERT INTO category (No, Name) VALUES
    -> (1, '计算机'),
    -> (2, '文学'),
    -> (3, '历史'),
    -> (4, '科学');
Query OK, 4 rows affected (0.013 sec)
Records: 4  Duplicates: 0  Warnings: 0

obclient(root@sys)[exp2]> INSERT INTO books (No, Title, Author, publication_year, C_no) VALUES
    -> (1, 'Python编程实战', '张三', 2021, 1),
    -> (2, '百年孤独', '加西亚·马尔克斯', 1967, 2),
    -> (3, '明朝那些事儿', '当年明月', 2006, 3),
    -> (4, '时间简史', '霍金', 1988, 4),
    -> (5, 'Java核心技术', 'K.S.霍斯特曼', 2019, 1);
Query OK, 5 rows affected (0.014 sec)
Records: 5  Duplicates: 0  Warnings: 0

obclient(root@sys)[exp2]> INSERT INTO readers (No, Name, Gender, Tel) VALUES
    -> (1, '小明', 'Male', '13800138000'),
    -> (2, '小红', 'Female', '13900139000'),
    -> (3, '小刚', 'Male', '13600136000'),
    -> (4, '小美', 'Female', '13700137000'),
    -> (5, '小强', 'Male', '13300133000');
Query OK, 5 rows affected (0.015 sec)
Records: 5  Duplicates: 0  Warnings: 0

obclient(root@sys)[exp2]> INSERT INTO borrow_records (No, B_no, R_no, Borrow_date, Return_date) VALUES
    -> (1, 1, 1, '2024-10-01', '2024-10-20'),
    -> (2, 2, 2, '2024-10-05', '2024-10-25'),
    -> (3, 1, 3, '2024-10-10', '2024-10-30'),
    -> (4, 3, 4, '2024-10-15', '2024-11-01'),
    -> (5, 2, 5, '2024-10-20', NULL);
Query OK, 5 rows affected (0.031 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

⑷使用比较运算符查询 2020 年以后出版的图书的 Title 和 publication_year

```
SELECT Title, publication_year
FROM books
WHERE publication_year > 2020;
```

```
obclient(root@sys)[exp2]> SELECT Title, publication_year
    -> FROM books
    -> WHERE publication_year > 2020;
+--------------------+------------------+
| Title              | publication_year |
+--------------------+------------------+
| Python编程实战 |             2021 |
+--------------------+------------------+
1 row in set (0.003 sec)
```

⑸查询借阅过编号为 3 的图书的读者的 Name 和 Tel

```
SELECT r.Name, r.Tel
FROM readers r
JOIN borrow_records br ON r.No = br.R_no
WHERE br.B_no = 3;
```

```
obclient(root@sys)[exp2]> SELECT r.Name, r.Tel
    -> FROM readers r
    -> JOIN borrow_records br ON r.No = br.R_no
    -> WHERE br.B_no = 3;
+--------+-------------+
| Name   | Tel         |
+--------+-------------+
| 小美 | 13700137000 |
+--------+-------------+
1 row in set (0.022 sec)
```

⑹查询作者姓名中包含“张”字的图书信息

```
SELECT *
FROM books
WHERE Author LIKE '%张%';
```

```
obclient(root@sys)[exp2]> SELECT *
    -> FROM books
    -> WHERE Author LIKE '%张%';
+------+--------------------+--------+------------------+------+
| No   | Title              | Author | publication_year | C_no |
+------+--------------------+--------+------------------+------+
|    1 | Python编程实战 | 张三 |             2021 |    1 |
+------+--------------------+--------+------------------+------+
1 row in set (0.002 sec)
```

⑺查询所有男性读者的借阅记录，包括读者姓名、借阅图书 Title、借阅日期和归还日期，结果按借阅日期升序排列。

```
SELECT r.Name, b.Title, br.Borrow_date, br.Return_date
FROM readers r
JOIN borrow_records br ON r.No = br.R_no
JOIN books b ON br.B_no = b.No
WHERE r.Gender = 'Male'
ORDER BY br.Borrow_date ASC;
```

```
obclient(root@sys)[exp2]> SELECT r.Name, b.Title, br.Borrow_date, br.Return_date
    -> FROM readers r
    -> JOIN borrow_records br ON r.No = br.R_no
    -> JOIN books b ON br.B_no = b.No
    -> WHERE r.Gender = 'Male'
    -> ORDER BY br.Borrow_date ASC;
+--------+--------------------+-------------+-------------+
| Name   | Title              | Borrow_date | Return_date |
+--------+--------------------+-------------+-------------+
| 小明 | Python编程实战 | 2024-10-01  | 2024-10-20  |
| 小刚 | Python编程实战 | 2024-10-10  | 2024-10-30  |
| 小强 | 百年孤独       | 2024-10-20  | NULL        |
+--------+--------------------+-------------+-------------+
3 rows in set (0.017 sec)
```

⑻查询女性读者的总人数

```
SELECT COUNT(*) AS Female_Count
FROM readers
WHERE Gender = 'Female';
```

```
obclient(root@sys)[exp2]> SELECT COUNT(*) AS Female_Count
    -> FROM readers
    -> WHERE Gender = 'Female';
+--------------+
| Female_Count |
+--------------+
|            2 |
+--------------+
1 row in set (0.001 sec)
```

⑼查询借阅时长超过 15 天的借阅记录，包括读者姓名、图书 Title

```
SELECT r.Name, b.Title
FROM readers r
JOIN borrow_records br ON r.No = br.R_no
JOIN books b ON br.B_no = b.No
WHERE DATEDIFF(br.Return_date, br.Borrow_date) > 15;
```

```
obclient(root@sys)[exp2]> SELECT r.Name, b.Title
    -> FROM readers r
    -> JOIN borrow_records br ON r.No = br.R_no
    -> JOIN books b ON br.B_no = b.No
    -> WHERE DATEDIFF(br.Return_date, br.Borrow_date) > 15;
+--------+--------------------+
| Name   | Title              |
+--------+--------------------+
| 小明 | Python编程实战 |
| 小刚 | Python编程实战 |
| 小红 | 百年孤独       |
| 小美 | 明朝那些事儿 |
+--------+--------------------+
4 rows in set (0.007 sec)
```

⑽向 borrow_records 表中插入一条记录（借阅编号为 6，图书编号为 5，读者编号为 5，借阅日期为'2025-03-24'，归还日期为空）。

```
INSERT INTO borrow_records (No, B_no, R_no, Borrow_date, Return_date)
VALUES (6, 5, 5, '2025-03-24', NULL);
```

```
obclient(root@sys)[exp2]> INSERT INTO borrow_records (No, B_no, R_no, Borrow_date, Return_date)
    -> VALUES (6, 5, 5, '2025-03-24', NULL);
Query OK, 1 row affected (0.004 sec)

obclient(root@sys)[exp2]> SELECT * FROM borrow_records;
+------+------+------+-------------+-------------+
| No   | B_no | R_no | Borrow_date | Return_date |
+------+------+------+-------------+-------------+
|    1 |    1 |    1 | 2024-10-01  | 2024-10-20  |
|    2 |    2 |    2 | 2024-10-05  | 2024-10-25  |
|    3 |    1 |    3 | 2024-10-10  | 2024-10-30  |
|    4 |    3 |    4 | 2024-10-15  | 2024-11-01  |
|    5 |    2 |    5 | 2024-10-20  | NULL        |
|    6 |    5 |    5 | 2025-03-24  | NULL        |
+------+------+------+-------------+-------------+
6 rows in set (0.001 sec)
```

⑾计算每本图书的平均借阅时长，并输出平均借阅时长超过 15 天的图书的 No

```
SELECT br.B_no
FROM borrow_records br
WHERE br.Return_date IS NOT NULL
GROUP BY br.B_no
```

HAVING AVG(DATEDIFF(br.Return_date, br.Borrow_date)) > 15;

```
obclient(root@sys)[exp2]> SELECT br.B_no
    -> FROM borrow_records br
    -> WHERE br.Return_date IS NOT NULL
    -> GROUP BY br.B_no
    -> HAVING AVG(DATEDIFF(br.Return_date, br.Borrow_date)) > 15;
+------+
| B_no |
+------+
|    1 |
|    2 |
|    3 |
+------+
3 rows in set (0.003 sec)
```

⑿查询借阅过编号为 1 或者编号为 2 图书的读者 No

SELECT DISTINCT R_no
FROM borrow_records
WHERE B_no = 1 OR B_no = 2;

```
obclient(root@sys)[exp2]> SELECT DISTINCT R_no
    -> FROM borrow_records
    -> WHERE B_no = 1 OR B_no = 2;
+------+
| R_no |
+------+
|    1 |
|    2 |
|    3 |
|    5 |
+------+
4 rows in set (0.002 sec)
```

⑬查询既借阅过编号为 1 又借阅过编号为 3 图书的读者 No

SELECT DISTINCT br1.R_no
FROM borrow_records br1
JOIN borrow_records br2 ON br1.R_no = br2.R_no
WHERE br1.B_no = 1 AND br2.B_no = 3;

```
obclient(root@sys)[exp2]> SELECT DISTINCT br1.R_no
    -> FROM borrow_records br1
    -> JOIN borrow_records br2 ON br1.R_no = br2.R_no
    -> WHERE br1.B_no = 1 AND br2.B_no = 3;
Empty set (0.003 sec)
```

⑭建立一个包含图书名字，作者，出版年份和图书类别的视图（赋予列名为 stitle, sauthor, spublicationyear, categoryname）。【create view】

CREATE VIEW book_details AS
SELECT b.Title AS stitle,
    b.Author AS sauthor,
    b.publication_year AS spublicationyear,
    c.Name AS categoryname
FROM books b

JOIN category c ON b.C_no = c.No;

```
obclient(root@sys)[exp2]> CREATE VIEW book_details AS
    -> SELECT b.Title AS stitle,
    ->        b.Author AS sauthor,
    ->        b.publication_year AS spublicationyear,
    ->        c.Name AS categoryname
    -> FROM books b
    -> JOIN category c ON b.C_no = c.No;
Query OK, 0 rows affected (0.127 sec)

obclient(root@sys)[exp2]> SELECT * FROM book_details;
+--------------------+------------------------+------------------+--------------+
| stitle             | sauthor                | spublicationyear | categoryname |
+--------------------+------------------------+------------------+--------------+
| Python编程实战 | 张三                   |             2021 | 计算机     |
| 百年孤独       | 加西亚·马尔克斯 |             1967 | 文学       |
| 明朝那些事儿 | 当年明月               |             2006 | 历史       |
| 时间简史       | 霍金                   |             1988 | 科学       |
| Java核心技术   | K.S.霍斯特曼       |             2019 | 计算机     |
+--------------------+------------------------+------------------+--------------+
5 rows in set (0.015 sec)
```

出现的问题：

⑴输出表格竖线无法对齐的问题

在查询后输出结果时，发现表中竖线无法对齐，以最后一题为例，在建好 view 后对其进行查询，发现表格竖线非常混乱且无法调整。

⑵数据类型设置为 VARCHAR 时会报错

在新建 table 时，第一次我将某字段的数据类型设为 VARCHAR，但表格创建无法成功，代码显示语法错误。

```
obclient(root@sys)[exp2]> CREATE TABLE category (
    ->     No INTEGER NOT NULL,
    ->     Name VARCHAR NOT NULL,
    ->     PRIMARY KEY (No)
    -> );
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
 to your OceanBase version for the right syntax to use near 'NOT NULL,
    PRIMARY KEY (No)
)' at line 3
[172.17.0.2:2882] [2025-04-15 09:06:52.554398] [YB42AC110002-000632CD0C48B469-0-0]
```

⑶创建 book 表时，提示“外键约束错误”

在新建 table 时，第一次我创建 books 表出现外键约束错误的报错。

解决方案：

⑴输出表格竖线无法对齐的问题的解决

经查询资料，这是因为中文字体宽度的问题，与实验结果正确与否无关，解决方法是输出为 txt 或换用其他可视化展示方式。下面是在 MySQL Workbench 8.0 CE 中的输出：

	stitle	sauthor	spublicationyear	categoryname
▶	Python编程实战	张三	2021	计算机
	Java核心技术	K.S.霍斯特曼	2019	计算机
	百年孤独	加西亚·马尔克斯	1967	文学
	明朝那些事儿	当年明月	2006	历史
	时间简史	霍金	1988	科学

⑵数据类型设置为 VARCHAR 时会报错问题的解决

经查询资料，在 Oceanbase 中新建 table 时若使用 VARCHAR 数据类型必须标清长度，我改为 VARCHAR(50)后重新创建就不会报错了。

⑶创建 book 表时，提示“外键约束错误”问题的解决

外键引用的表（如 category）尚未创建，创建 category 后再创建 books 就不会报错了。