



# Reel Synapse

影碟零售中台系统



CAINE · XU





# Introduction

## 童年记忆的数字化转型

基于C#与SQL Server的桌面应用系统，实现影碟管理、会员管理、销售与退货管理等核心业务流程的信息化与自动化。

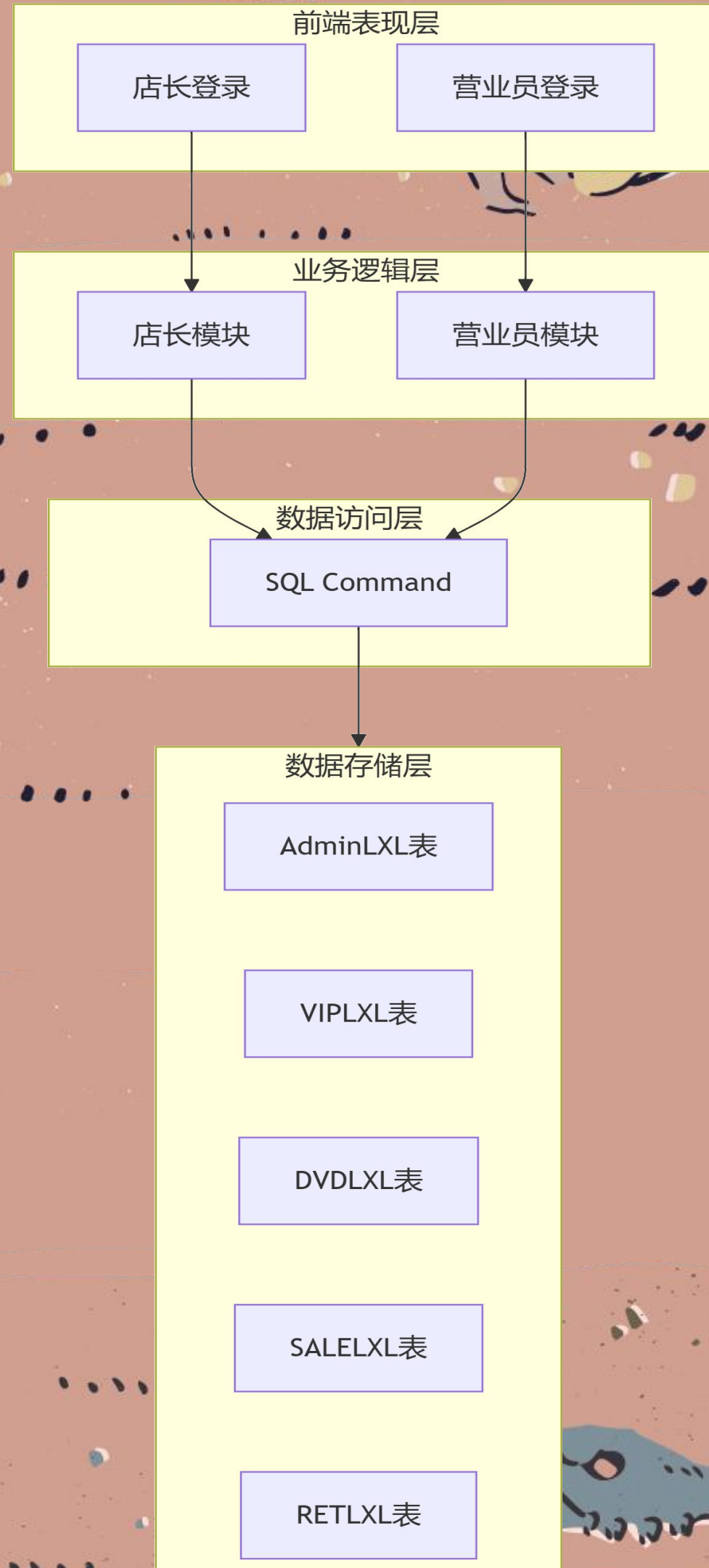
核心亮点：严谨的数据库设计与优化；事务性与业务闭环设计；注重用户体验的业务逻辑。

产品价值：

效率提升：替代传统手工记账与人工盘点，实现影碟库存、会员信息、销售账单的数字化管理。

体验优化：支持会员制购物、快速退货、账单追溯，提升顾客满意度。

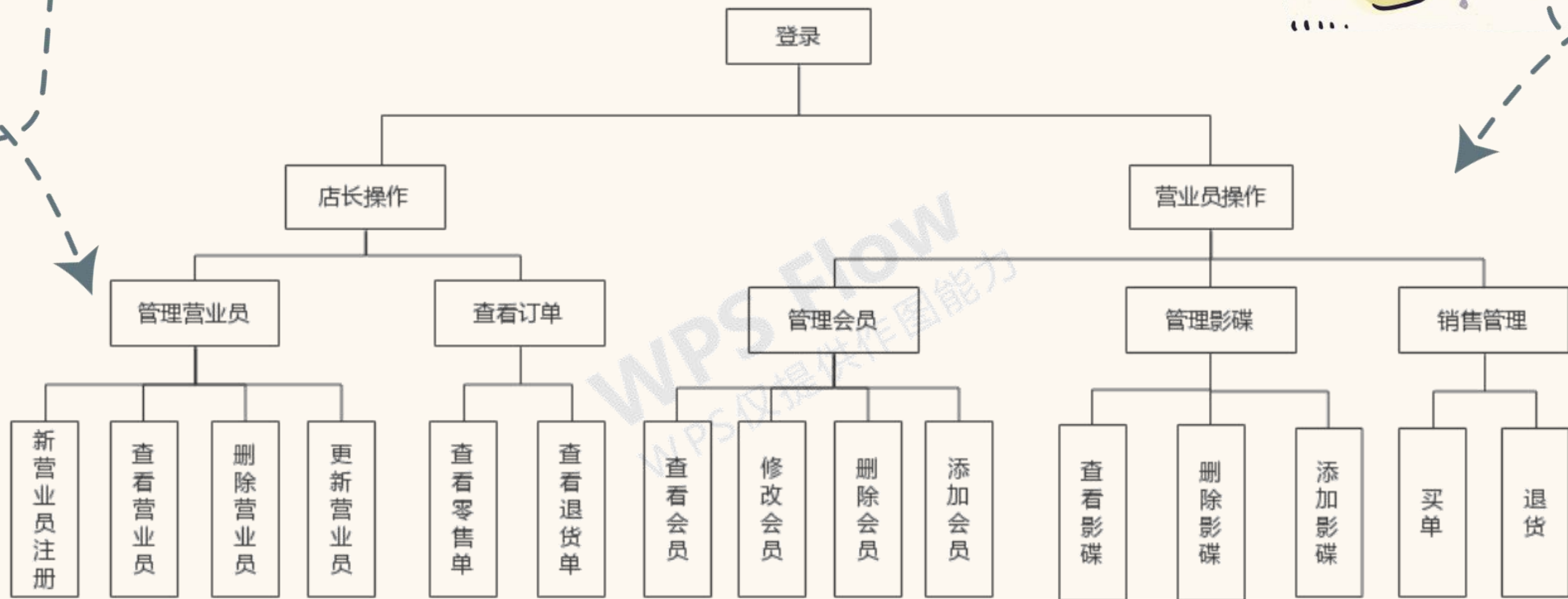
# Tech Strack



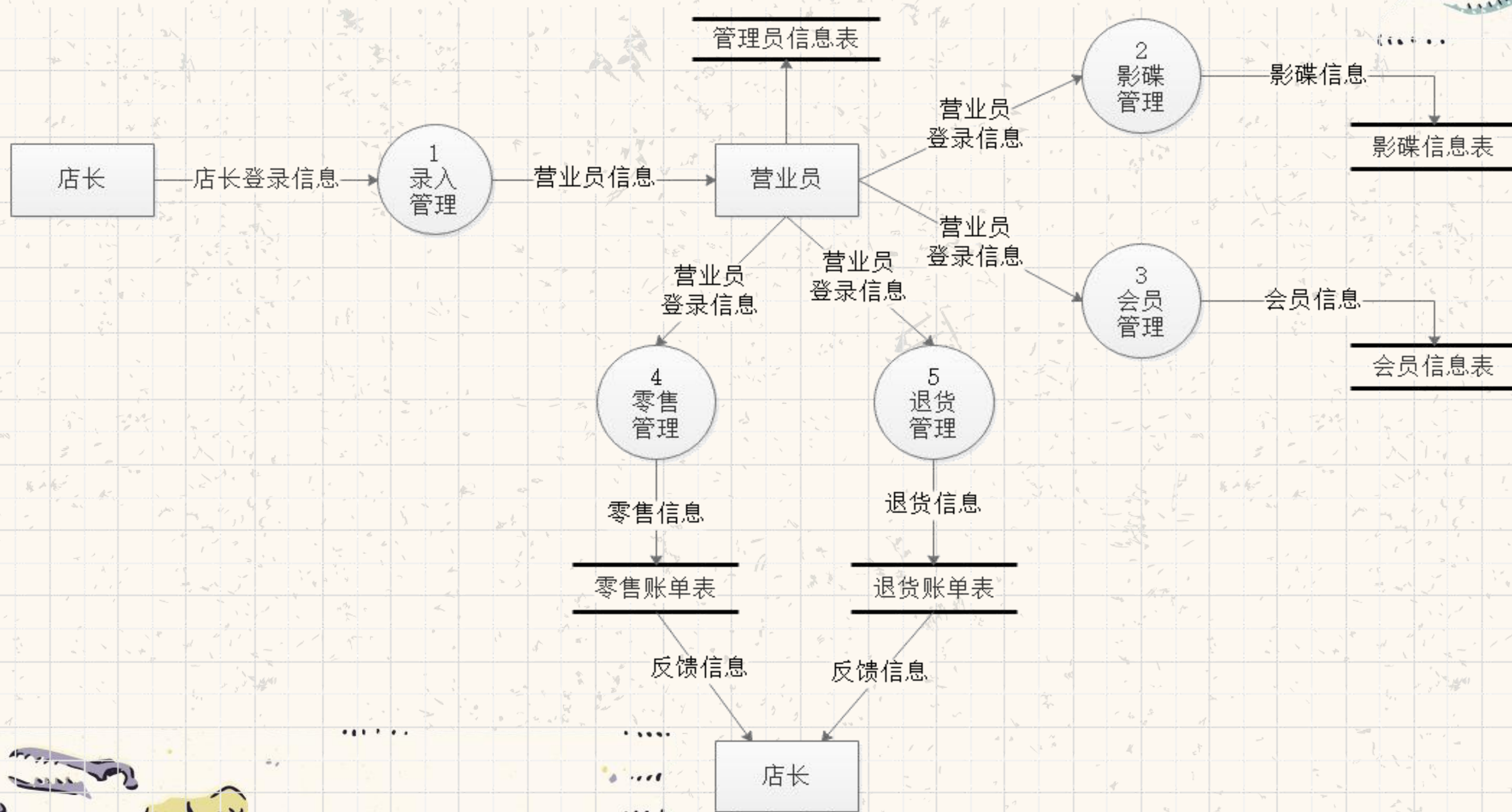
- 三层架构 (UI, Logic, Data Access)
- 前端: C#, .NET Framework构建用户界面, 支持店长与营业员分级登录与操作
- 后端: SQL Server 作为数据存储层, 通过ADO.NET实现数据连接与事务处理
- 数据库连接: SqlConnection 实现本地数据库连接;
- 登录验证: 根据角色分别查询不同表;
- 索引优化: 为影碟名称、会员电话等高频查询字段建立索引。
- 权限控制: 通过数据库角色public限制对表的操作, 确保数据安全。



# Function Structure

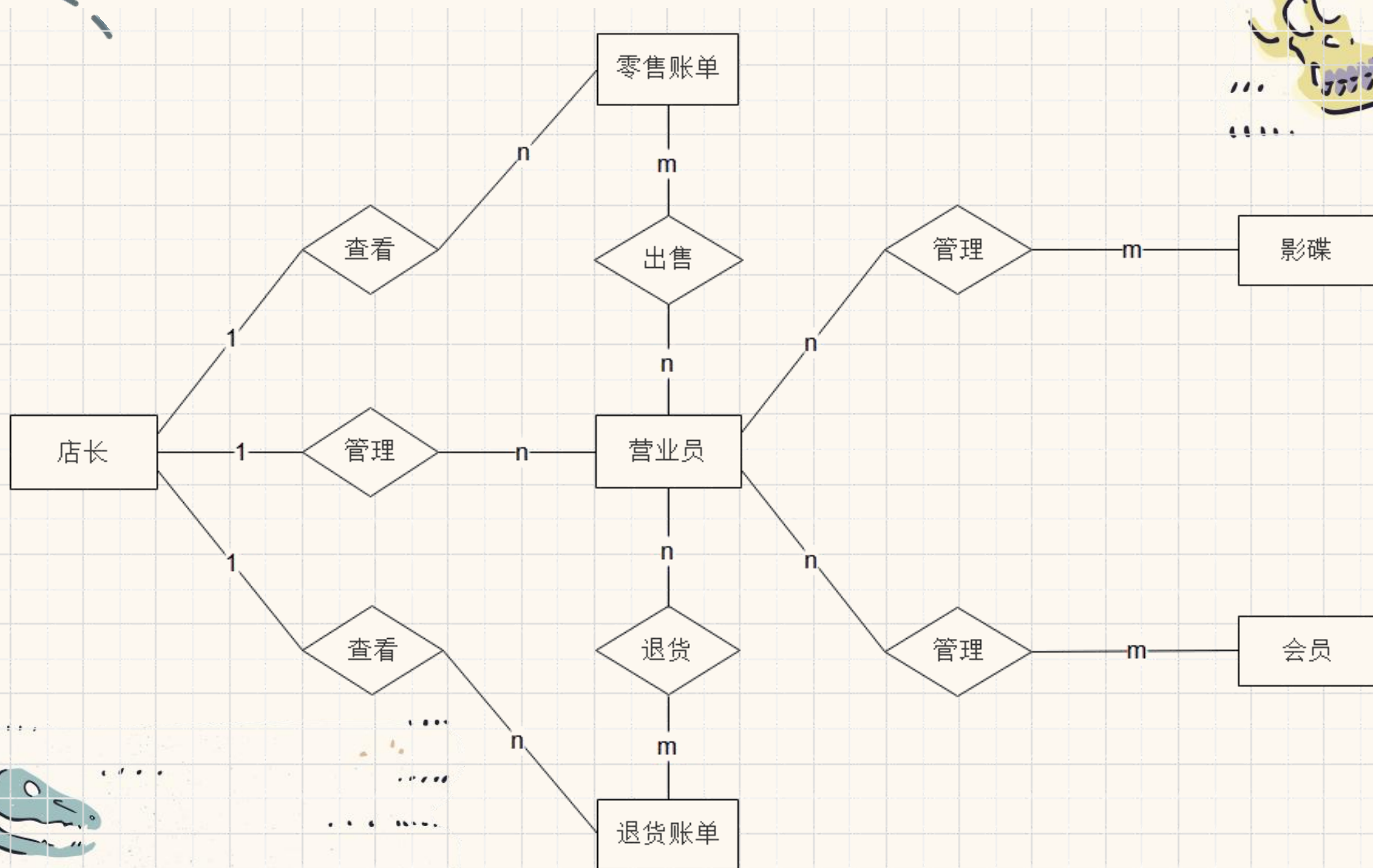


# Data Flow Diagram





# Overall E-R Diagram



# Critical Function and Key Code

碟片出售

SQL CODING

```
CREATE PROCEDURE insert_sbill
    @dvdid INT, @buynumb INT,
    @vipid INT, @adid CHAR(20)
AS
BEGIN
    DECLARE @price INT = 15 *
    @buynumb; -- 计算金额
    -- 插入新订单
    INSERT INTO SALELXL(saledvdid,
        salenumb, saleprice, salevip,
        saleadmin)
        VALUES(@dvdid, @buynumb,
            @price, @vipid, @adid);
    -- 更新影碟库存
    UPDATE DVDLXL SET DvdNumb =
        DvdNumb - @buynumb WHERE
        DvdID = @dvdid;
END
GO
```

鸭卖碟片管理系统—营业员卖碟

每张影碟定价为十五元人民币! ~

购买碟片 退货退款

	影碟编号	影片名称	影片库存
*			

查 询

☒ 所 有 ☐ 片 名 ☐ 语 言 ☐ 类 型

查 询

请确认购买的碟片信息

碟片编号

购买数量

会员联系方式

确 认 购 买

返 回 退 出 (E)



# Critical Function and Key Code

碟片出售 C# CODING

## ① 查询碟片编号

```
string ifexist = "select * from DVDLXL where DvdID = '"+buydvdid+"'";  
SqlCommand ifex = new SqlCommand(ifexist, conn);  
SqlDataReader d = ifex.ExecuteReader();
```

## ② 查询影碟库存是否足够

```
string checkdvd = "select * from DVDLXL where DvdID = '" + id + "' and DvdNumb >= 0 ";//是否还有库存
```

有库存

```
string checkdvdb = "select * from DVDLXL where DvdID = '" + id + "' and DvdNumb >= '" + numb + "'";//是否有足够库存
```

## ③ 插入新的订单，插入订单后修改影碟库存

```
string bill = "EXEC insert_sbill @dvdid = '" + id + "',@buynumb = '" + numb + "',@vipid = ",@adid  
= '" + db_Login.username + "'";
```

```
SqlCommand sql1 = new SqlCommand(bill, conn);//插入账单  
sql1.ExecuteNonQuery();//进行操作
```

```
string salebill = "EXEC Select_sbill";//选择最新的一条 id
```

```
string updatedvd = "EXEC updatedvd";//更新 dvd 库存
```

```
SqlCommand sql2 = new SqlCommand(salebill, conn);
```

```
SqlCommand sql3 = new SqlCommand(updatedvd, conn);
```

```
sql3.ExecuteNonQuery();//
```





# Critical Function and Key Code

退货管理

CODING

鸭卖碟片管理系统—营业员卖碟

每张影碟定价为十五元人民币! ~

购买碟片 退货退款

查 询

☒ 所 有 ☐ 出售日期 ☐ 营业员工号 ☐ 会员ID

查 询

输入购买单号

退 货

返 回

```
CREATE PROCEDURE INSERT_retlxl
@retsale INT
AS
DECLARE @i INT , @id INT,@num INT
INSERT INTO RETLXL (Retsaleid,Retprice)
select saleid,saleprice*2/3 from SALELXL
WHERE saleid = @retsale
UPDATE SALELXL SET salestu = 1 where
saleid = @retsale
SELECT TOP 1 @i = Retsaleid FROM
RETLXL ORDER BY Retid DESC
SELECT @id = saledvdid, @num =
salenumb FROM SALELXL WHERE saleid
= @i
UPDATE DVDLXL SET DvdNumb =
DvdNumb+@num WHERE DvdID = @id
GO
```



# Critical Function and Key Code

碟片出售 C# CODING

```
string selsaleid = "select * from SALELXL where salestu = 0";
```

```
SqlCommand sql = new SqlCommand(selsaleid, conn);
```

```
SqlDataReader dr = sql.ExecuteReader();
```

```
string insertret = "EXEC INSERT_ret1xl @retsale='" + THDid + "'"; //新建退货单
```

```
SqlCommand sql1 = new SqlCommand(insertret, conn);
```

```
sql1.ExecuteNonQuery();
```

```
string sq2 = "EXEC Select_retbill"; //查看
```

```
SqlCommand sql2 = new SqlCommand(sq2, conn);
```

```
SqlDataReader dr2 = sql2.ExecuteReader();
```





# Improvement



## 全栈开发能力

独立完成了从数据库设计（SQL Server）、后端业务逻辑（C#）到前端界面（WinForms）的全链路开发，建立了完整的全栈知识体系。



## 产品与业务思维

不再仅仅关注代码实现，而是学会了如何进行需求分析、模块划分和用户体验设计，理解了技术最终是为业务价值服务的。



## 工程化与调试能力

通过处理多用户并发、数据一致性等复杂场景，以及大量的功能测试，系统化地提升了我的软件调试和稳健性保障能力。

# Reflection

ReelSynapse 是我第一次以产品负责人的视角去完成一个完整的系统。

从最初模糊的“影碟店管理”需求，到最终一个功能完备、界面清晰的管理系统，我深刻体会到将用户需求精准转化为技术方案的重要性。

为了设计出符合店长和营业员直觉的交互流程，我特意去了解了实体零售店的真实工作模式。

这个过程让我明白，一个优秀的产品不能只停留在技术实现，更需要深入到业务场景中去。