

# 数据库系统课程设计：智慧物流车队与配送管理系统

## 一、 数据库语义规则

为了构建符合现实逻辑的数据库，定义以下语义规则：

1. **配送中心与车队**：物流公司设有若干个区域配送中心，每个中心下辖若干个运输车队（例如：干线车队、同城配送车队、A队、B队、[自定义名字]队）。
2. **车辆管理**：每个车队拥有若干辆运输车，每辆车有固定的**最大载重**和**最大容积**；每辆车只能属于一个车队。
3. **人员架构**：
  - 每个车队有一名**调度主管**，只负责排班和车辆调度，不参与驾驶。
  - 每个车队有若干名司机
4. **运单与任务**：
  - 系统中有流转的“运单”（Order），包含货物重量、体积、目的地等信息。
  - 运单会被分配给某辆车进行运输。
5. **动态状态与异常**：
  - 车辆在运输过程中会产生状态变更（如：装货中、运输中、已签收）。
  - 运输过程中可能发生“异常事件”，包括：货物破损、车辆故障、严重延误、超速报警等。
6. **信息录入权限**：调度主管可以对自己监管的所有车辆进行查询、录入和更改异常信息和运单分配信息，司机只能查询自己的信息。

## 二、 系统开发要求

开发一个管理系统对数据库进行访问，前端可使用Java（SpringBoot/Swing），Python（Django/Flask/PyQt），C#等技术栈。系统需实现以下功能，必须提供图形界面完成以下操作，图形界面无需美化，能够体现出业务逻辑即可，禁止纯SQL命令行：

1. **基础信息管理**：录入司机（工号、驾照等级、联系方式等）和车辆（车牌、载重上限、容积上限、当前状态）的基本信息。[前端表单需做基础校验（非空、驾照等级枚举值、车牌正则）]。
2. **运单分配**：将新的运单分配给指定的车辆。[下拉列表或搜索框选择“空闲且载重足够”的车辆，超载时后端触发器抛异常→前端捕获并弹窗提示“超出最大载重”。]
3. **异常记录录入**：录入运输过程中的异常事件（关联司机、车辆、时间、异常类型（运输中异常、空闲时异常）、罚款金额等），以及异常处理状态（已处理/未处理）。

4. **车队资源查询**：查询某个配送中心下所有车队的车辆负载情况，如：哪些车空闲，哪些车暂时无法运输（满载或异常的车）。
5. **司机绩效追踪**：查询某名司机在特定时间段内的运输单数及产生的异常记录详情。
6. **统计报表**：查询某个车队在某个月度的“安全与效率报表”，包含：总运单数、异常事件总数、累计罚款金额。

### 三、数据库设计重点注意事项

1. **规范化设计**：严格遵循数据库规范化理论（至少达到3NF），消除数据冗余，避免更新异常。
2. **完整性约束**：
  - **实体完整性**：正确设置主键。
  - **参照完整性**：正确设置外键关联（如：车辆属于车队，运单属于车辆）。
  - **用户自定义完整性**：例如，车辆状态只能是“空闲”、“运输中”、“维修中”。
3. **高级数据库对象的使用**：
  - 必须使用触发器（Trigger）实现以下逻辑：
    - **自动载重校验**：当向一辆车分配运单时，触发器自动检查该车“当前已分配货物重量 + 新运单重量”是否超过“车辆最大载重”。如果超过，则**拒绝插入并抛出错误**。
    - **车辆状态自动流转**：当一辆车完成所有运单的签收（运单状态全部变为“已完成”），触发器自动将车辆状态从“运输中”更新为“空闲”。当一辆处于异常状态的车的异常处理完成后，触发器自动将车辆状态从“异常”更新为“空闲”或“运输中”，取决于异常类型。
    - **审计日志**：当修改司机的关键信息（如驾照等级）或异常记录被处理时，触发器自动将旧数据写入一张独立的 **History\_Log** 表中进行备份。
    - **存储过程**：编写存储过程用于复杂的统计计算（如计算车队月度绩效）。
    - **视图**：建立视图以简化查询（如创建一个“车队异常警报视图”，只显示本周发生过异常的车辆和司机信息）。
  - 4. **索引优化**：对高频查询字段（如车牌号、运单日期、司机工号）建立索引，并对比建立索引前后的查询效率（可用 **Explain** 分析）。
  - 5. **工具使用**：使用PowerDesigner、DataGrip或Draw.io等绘图软件绘制规范的E-R图和数据模型图。

### 四、作业报告撰写大纲

请按照以下结构撰写课程设计报告：

0. 成员
  - 小组成员

- 成员分工总览，在后面实验报告的具体部分也要相应注明

## 1. 概念结构设计

- 实体与属性定义
- E-R图（需清晰标注 1:1, 1:n, m:n 关系）

## 2. 逻辑结构设计

- E-R图转换为关系模式（表格结构）
- 规范化分析（证明至少已达到3NF）

## 3. 物理结构与高级对象设计（重点）

- 表结构定义（SQL建表语句）
- Trigger的详细设计与代码实现（需解释设计意图）
- 存储过程与视图的设计
- 索引的设置策略

## 4. 系统实现与测试

- 开发环境说明
- 关键功能截图（特别是触发器生效拦截错误或自动更新数据的截图）
- 前端技术选型与关键代码
- 遇到的一处具体技术难点及解决方案

## 5. 总结

- 课程设计心得
- 对数据库效率优化的思考

## 6. 附录/附件

- 主要SQL脚本（建表、触发器、存储过程）
- 演示demo，录制以小组成员学号（口述）开始的演示作业设计中各个功能的简短视频,如果不方便传视频，可以传到一些在线视频网站然后把链接附在报告中，助教由视频开头的口述学号确定视频归属者。
- 前端代码不要求上传，但需要在视频中体现。

# 五、 实验安排

1. **分组**：三人一组完成实验任务，实验报告按照小组提交，需写明每位组员的完成部分以及贡献百分比以便评分（严禁出现xx完成实验，xx撰写报告这样分工不明的）若有特殊情况请联系助教。

2. 提交链接：<https://yunbiz.wps.cn/c/collect/cBDmwX4zZmK>

3. 分组时间：2025.12.28前

- 4. 报告提交截至时间： 2025年1月17日23:59**
- 5. 分工举例：**数据库设计（25%）； 触发器/存储过程（25%）； 后端接口（25%）； 单元测试（25%）； 简单前端（25%）