**课 程 设 计 报 告**

|  |  |
| --- | --- |
| 学 院： |  |
| 专业名称： |  |
| 课程名称： |  |
| 课程代码： |  |
| 所在班级： | 2020级4班 |
| 学 号： |  |
| 姓 名： | 周鹏鹏 |
| 课题名称： |  |
| （分题目） | 骑手人事管理系统——招聘管理 |
| 任课教师： |  |
| 完成时间： | 2021年10月 - 2022年12月 |

**数据库课程设计成绩评定表**

|  |  |  |  |
| --- | --- | --- | --- |
| **报告题目** |  | | |
| **评价内容** | **标 准** | **分值** | **得分** |
| 1．选题 | 课题能体现本课程的培养目标和要求，课题难度和设计实现的工作量适中，有一定的创新性。 | 10 |  |
| 2．设计与实现 | 功能模型及模块划分科学完整、层次清楚。概念模型的设计正确，DFD、E-R图正确。逻辑模型的设计正确、关系模式论证正确。物理结构的设计正确，键字结构清楚合理。实现设计文理通顺，层次清楚。软件运行完成且正确，预计功能均顺利实现。 | 30 |  |
| 3．文档质量 | 报告设计科学，计算准确；图、表绘制及技术要求符合国家标准规范。结构严谨，层次分明，论述清晰，语言流畅，表达准确。课程设计报告格式规范。 | 30 |  |
| 4．陈述 | 可根据此设计文档正确陈述设计。思维敏捷、逻辑性强；回答问题有理有据，体现出有充分的准备。陈述清晰、正确；体现基本概念清楚，语言流畅、精炼。 | 10 |  |
| 5．协作及表现 | 在整个课程设计中能够独立分析并解决问题，完成相关部分的设计。每组完成一个共同的项目，其实现及报告的关联性及整体性体现了团队协作，表现较高的团队协作能力。 | 20 |  |
| 总 分 | | |  |

### 骑手人事管理系统设计与开发

目录

一．设计目的及要求**4**

1.1选题原因4

1.2课程设计目的5

1.3课程设计内容5

二．需求分析**5**

2.1分组及开发环境5

2.2系统规划6

2.2.1功能设计6

2.2.2可行性分析8

2.3系统分析8

2.3.1系统需求说明书 8

三．概念模式设计**12**

3.1数据抽象12

3.2局部E-R图设计13

3.3总体E-R图设计14

四．逻辑模式设计**14**

4.1数据逻辑结构14

4.2对各个数据类型的解释16

五．具体实现**17**

5.1数据库及表的创建17

5.2存储过程及约束设计18

5.3 C++配置ODBC本地连接数据库21

5.4界面创建25

5.4.1登录界面设计25

5.4.2查询修改界面设计27

5.4.3添加骑手界面设计29

5.4.4骑手招聘信息界面设计29

5.45骑手删除信息界面设计30

六.总结31

1. 设计目的及要求

1.1选题原因

外卖现在已经成为了人们的主流生活方式之一，现在人们的生活中不管是美食，商超，鲜花还是药品，打开手机APP下单，等着外卖小哥送货上门，这已经是千万都市人群的常态。

随着外卖市场的不断发展，外卖给人们生活方式带来了翻天覆地的改变，“懒人经济”成为一种现象，不管对我们的商家，用户还是骑手都带来了不可逆转的影响。

首先，外卖骑手作为新兴职业，为社会提供了更多的就业机会。据统计，2019年有398.7万名外卖骑手通过美团平台就业获得收入，比2018年的270多万人，增加了128.7万。并且，截至2019年底，累计已经约有720万名外卖骑手通过美团平台实现就业增收，有效带动社会就业。

外卖骑手这个职业对他们来说，不仅是稳定的收入来源，更是他们融入城市，打开、提升职业道路的通道。他们大多数来自农村，没有学历和技能，需要城市为他们提供更多的就业机会。而外卖骑手职业可以成为他们走进城市的第一站，帮助他们掌握基本的职业技能，畅通职业发展通道，使他们在异乡的漂泊城市中立足，获得别人的肯定和认可，共同为社会添砖增瓦。

除此之外，外卖骑手职业也是共享经济时代下的新就业方式，具有灵活就业特点，很大程度上满足了各类群体的就业需求。据调查，有不少工厂工人、企事业单位、自己做生意或创业、厨师、服务员、快递员、保安等其他工作人员都兼职做外卖骑手。这一方面有助于增加他们的收入，另一方面也有助于合理调动和配置劳动力资源。

同时，外卖骑手也为城市提供更多应急服务，带来很多社会价值。在疫情的特殊时期，大多数城市家庭都宅在家中，这些外卖小哥被人们尊重地称为护“胃”队、摆渡人，他们不仅送餐，也送药、送口罩、送油，保障了人们的生活物资，也为医疗一线提供急需的医疗物资，他们是真正的“英雄骑士”。

对于一名专业的信息技术人员来说，如何最大限度地利用市场后勤部门多年来在数据库系统上的投资，把市场花了大力气搜集来的数据变成对市场未来发展有巨大价值的信息，这是每个从事信息技术工作的专业人员都应该考虑的问题。而为了将人力资源工作者从繁重的日常琐碎事务中解放出来，将更多地精力用于外卖企业的人力资源职能管理和管理决策，保持企业的持续高效运营。 骑手人事管理系统应运而生，本系统面向于小型企业开发，以方便人员管理。

1．2课程设计目的

通过设计一个管理信息系统,考察学生运用所学知识的能力,通过对所学相关课程（计算机语言、数据结构、软件工程、数据库原理）知识的综合运用来提高学生的实践能力、综合知识应用能力，为毕业设计打下良好的基础。

1.3课程设计要求

1．知识与能力要求

在对数据库理论及知识理解的基础上；重点是针对具体的实际问题选择并设计合适的数据库表加以应用，并在此基础上完成相关的算法与程序；

2.内容的要求

（1）每人要充分认识课程设计对培养自己的重要性，认真做好设计前的各项准备工作。

（2）既要虚心接受老师的指导，又要充分发挥主观能动性。结合课题，独立思考，努力钻研，勤于实践，勇于创新。

（3）在设计过程中，要严格要求自己，树立严肃、严密、严谨的科学态度，必须按时、按质、按量完成课程设计。

1. 需求分析

2.1分组及开发环境

分组：

本小组由4个人组成：周鹏鹏（组长），谢鹏飞（组员），汪辛吉（组员），周再航（组员）

（图2.1）

开发环境：

(1)机房

(2)软、硬件

操作系统:Windows10

编程语言:C++ T-sql

编译器：MinGw32

辅助工具:1）文字处理—WORD 2）图片绘制—2012Visio,2012WPS 3）界面设计—Qt Creator

（3）数据库:SQL Server 2019

SQLServer tools: Microsoft SQL Server Management Studio 18

2.2系统规划：

2.2.1功能设计

**总系统功能设计：**

1、骑手人事管理系统下的招聘管理子系统，需要编辑发布招聘信息，筛选简历，安排面试，通过面试结果通知骑手办理培训入职等事项。过程抽象为：

**1.1 编辑发布招聘信息  
1.2 简历面试审核  
1.3 记录审核结果  
1.4 联系入职记录**

2、骑手人事管理系统下的入职管理子系统，需要收取员工资料，录入员工工作日程，记录员工培训结果，安排员工工具分配等事项。过程抽象为：  
 **2.1 收取员工资料  
 2.2 员工工作日程记录  
 2.3 员工培训结果记录  
 2.4 员工工具分配情况记录劳动合同管理**

3.骑手人事管理系统下的劳动合同管理子系统，需要通知通过审核的员工来签订合同，利用员工的工作时长签订不同的合同，接收来自离职部的员工的信息，然后通知员工签订更改后的合同等事项。过程抽象为：

**3.1 签订合同**

**3.2 分类签订合同的员工**

**3.3 劳动变更**

4、骑手人事管理系统下的离职管理子系统，需要接收离职信息，变更劳动合同，结算工资，离职交接等事项。过程抽象为：

**4.1 分类离职员工**

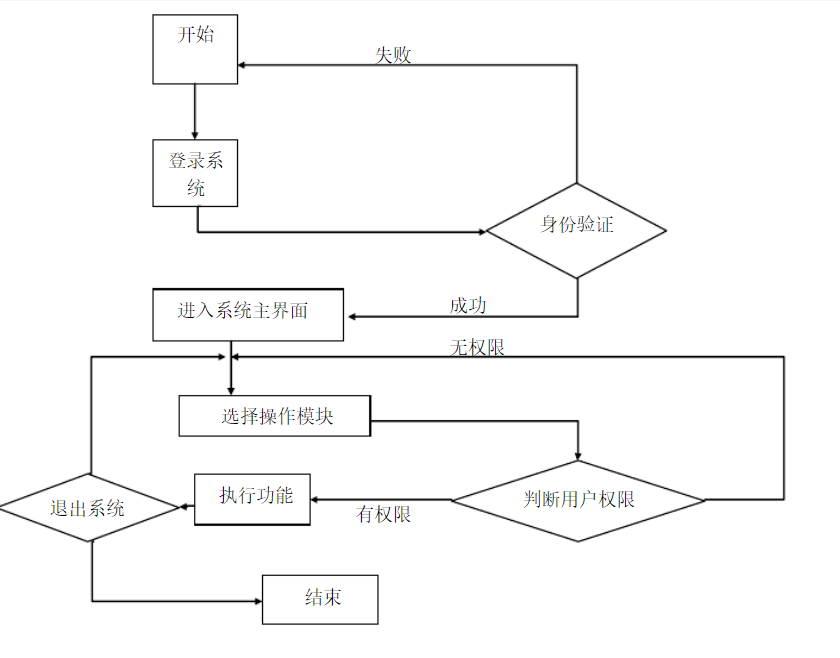
**4.2 填写离职信息**

**4.3 审批离职信息**

**4.4 删除员工信息**

图中的选择操作模块分为：**招聘管理操作模块**，**入职管理操作模块**，**劳动合同管理操作模块**，**离职管理操作模块**。

系统功能流程图：（设操作功能模块为**招聘管理操作模块**）

（图2.2）（图2.3）

招聘管理操作模块

F1.发布招聘信息

F2.记录审核结果

F3.骑手联系信息

F4.记录骑手报道结果

**而招聘管理操作模块下的主要功能分为：（招聘数据流图已标出）**

F1.编辑发布招聘信息

F2.记录审核结果

F3.汇总合格骑手联系信息

F4.记录骑手报道结果

2.2.2可行性分析

传统人工的方式管理文件档案存在着许多缺点，如效率低、保密性差等。另外时间一长，将产生大量的文件和数据，这对于查找、更新和维护都带来了不少的困难。

1)技术上的可行性

 此次系统开发使用C++语言和T-SQL，这些语言抽象能力高，执行效率快捷。再加上软件工程，数据库系统的理论基础知识，以及构建实践结构的工具：1）文字处理—WORD 2）图片绘制—2012Visio,2012WPS 3）能够让设计思路变得非常清晰，这些适合新手上路的软件完全能够支持开发一个骑手人事管理系统

2)管理上的可行性

因为系统主要应用于学生实践，而不具备网络查找功能，因此在实现时可以做成单机版，对管理员赋予数据录入、查询、修改以及对数据库的清零等功能，没有管理上的困难。

3)安全上的可行性

在建立数据库时可以通过对数据库的设计用户名和密码进行保密，以及在建立数据库以后可以对数据进行压缩等技术，保证数据的安全，使数据库具有安全保障。

4)社会因素的分析

随着科学技术的不断提高，计算机科学日渐成熟，作为计算机应用的一部分，使用计算机对学籍信息进行管理，具有着手工管理所无法比拟的优点。这些优点能够极大提高学生档案管理的效率，也是企业的科学化、正规化管理，与世界接轨的重要条件。该软件的开发不会侵犯国家、集体和他人的利益。

2.3系统分析

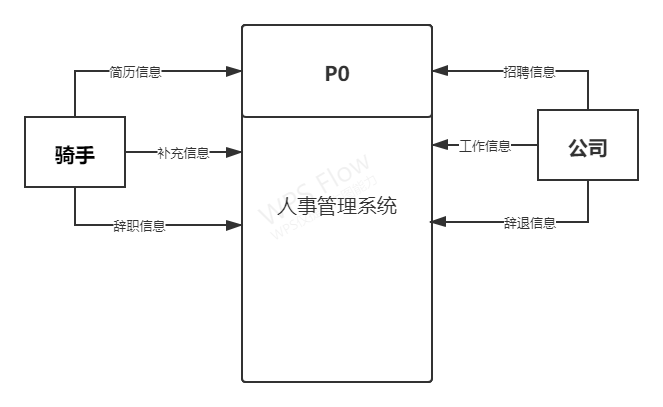
2.3.1系统需求说明书

数据流图设计：

人事管理系统（总体）：

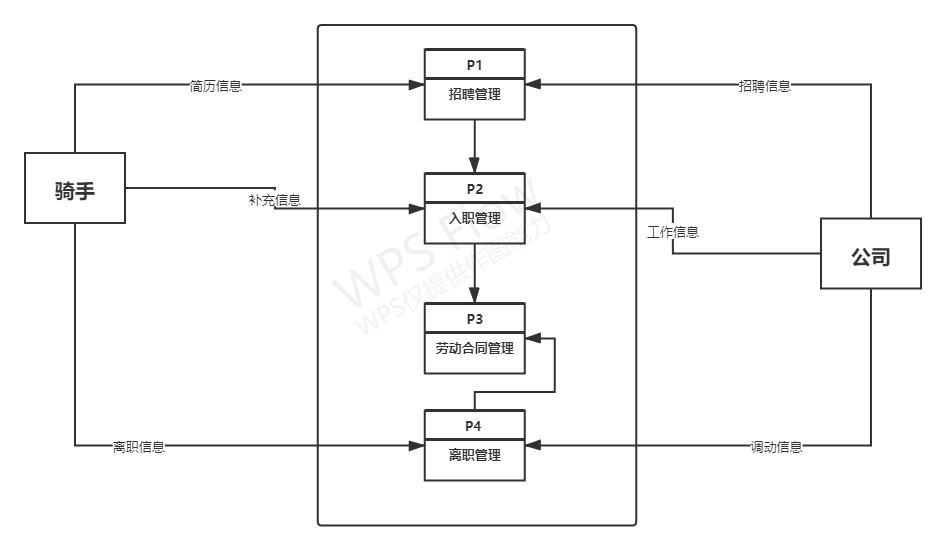
根据需求得到建立数据流图：

简化数据流图顶层：把整个系统视为 一个大的加工 (也只能含一个加工)，然后根据数据系统从哪些外部实体接收数据流，以及系统发送数据流到那些外部实体，就可以画出输入输出图。这张图称为顶层图。

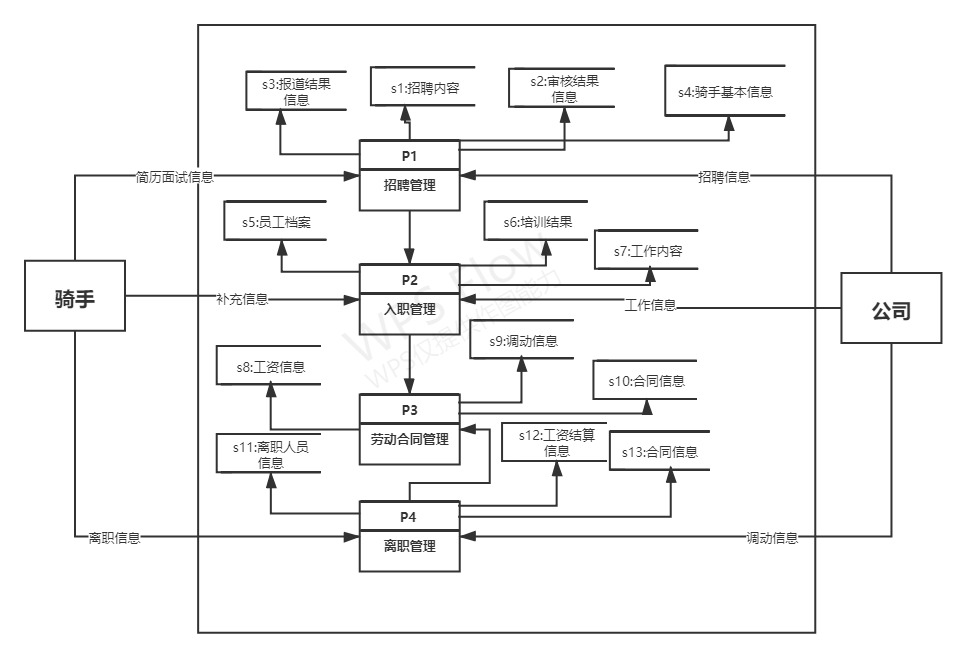
****（图2.4）

简化数据流图0层：

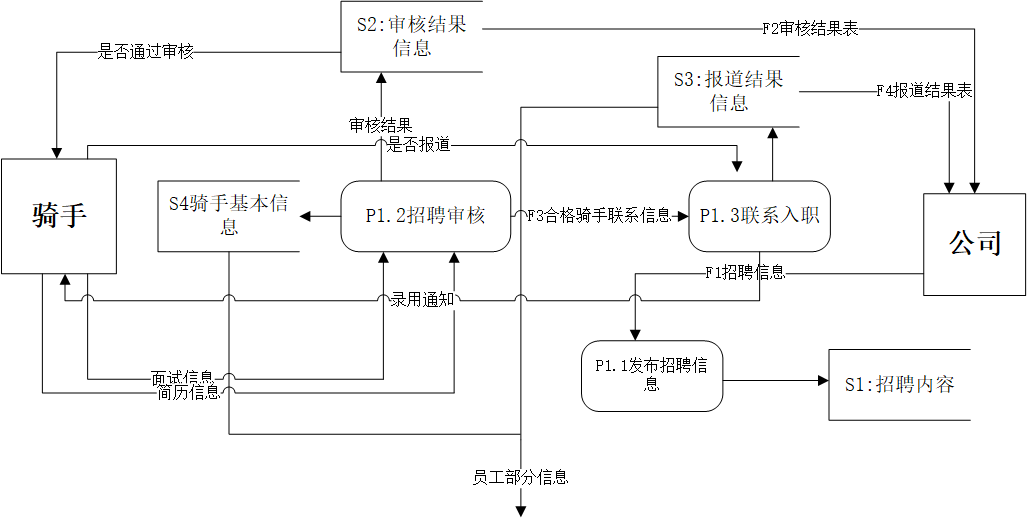
把顶层图的加工分解成若干个加工，并用数据流将这些加工连接起来，使得顶层图的输入数据经过若干加工处理后，变成顶层图的输出数据流。这张图称为0层图。从一个加工画出一张数据流图的过程就是对加工的分解。

****（图2.5）

自定义信息全貌图：

（图2.6）

招聘管理数据流图1层：

 （图2.7）数据流图及数据字典设计要点:

该子系统用户为各个外卖公司的骑手HR

数据存储表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 数据文件 | 文件组成 | 关键标识 | 组织 |
| 1 | S1招聘内容 | 年龄+薪资+工作地点+社保+招聘编号+面试时间+面试地点 | 招聘编号 |  |
| 2 | S2审核结果信息 | 应聘者编号+抗压指数+经验指数+交通意识指数+平均指数+是否录用 | 应聘者编号 | 是否录用+平均指数降序 |
| 3 | S3报道结果信息 | 应聘者编号+是否报道+联系方式 | 应聘者编号 | 是否报道降序，应聘者编号升序 |
| 4 | S4骑手基本信息 | 姓名+性别+年龄+联系方式+学历+应聘者编号+经验 | 应聘者编号 | 应聘者编号升序 |

**处理过程逻辑：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 处理过程 | 编号 | 输入 | 输出 | 处理逻辑 |
| 1 | 发布招聘信息 | P1.1 | 招聘内容 | 招聘信息 | 根据公司用人需求发布通知 |
| 2 | 招聘审核 | P1.2 | 面试信息+简历信息 | 平均指数+是否录用 | 平均指数是否过审 |
| 3 | 联系入职 | P1.3 | 合格骑手的联系信息 | 报道信息 | 联系骑手入职 |

数据项：

**各个数据项（数据构成改为正则表达式）**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 数据项 | 数据构成 | 数据对象说明 |
| 1.1 | 招聘编号 | ‘[100001-109999]’ | 1开头的6位数 |
| 1.2 | 年龄 | ‘[16-99]-[16-99]’ | 年龄范围/或者用最高,最低年龄表示 |
| 1.3 | 薪资 | ‘[0-3000]-[3000-12000]’ | 薪资范围/或者用最高,最低年龄表示 |
| 1.4 | 工作地点 | ‘%市%’ |  |
| 1.5 | 社保 | ‘%’ |  |
| 1.6 | 面试时间 | ‘%年[1-12]月’ |  |
| 1.7 | 面试地点 | ‘%’ |  |
| 2.1 | 抗压指数 | ‘[1-100]’ | 分数 |
| 2.2 | 经验指数 | ‘[1-100]’ | 分数 |
| 2.3 | 交通意识指数 | ‘[1-100]’ | 分数 |
| 2.4 | 平均指数 | ‘[1-100]’ | 前三个数据求平均 |
| 2.5 | 是否录用 | ‘[0,1]’ | 0否，1是 |
| 3.1 | 是否报道 | ‘[0,1]’ | 0否，1是 |
| 4.1 | 应聘者编号 | ‘[100001-109999]’ | 主键唯一 |
| 4.2 | 姓名 | ‘%’ |  |
| 4.3 | 性别 | ‘[0,1]’ | 0为女，1为男 |
| 4.4 | 年龄 | ‘[16-99]’ |  |
| 4.5 | 联系电话 | ‘1[0-9]{10}’ | 1开头的11位数字 |
| 4.6 | 学历 | ‘%’ |  |
| 4.7 | 经验 | ‘%’ |  |

三．概念模式设计

3.1数据抽象

两种常用抽象：

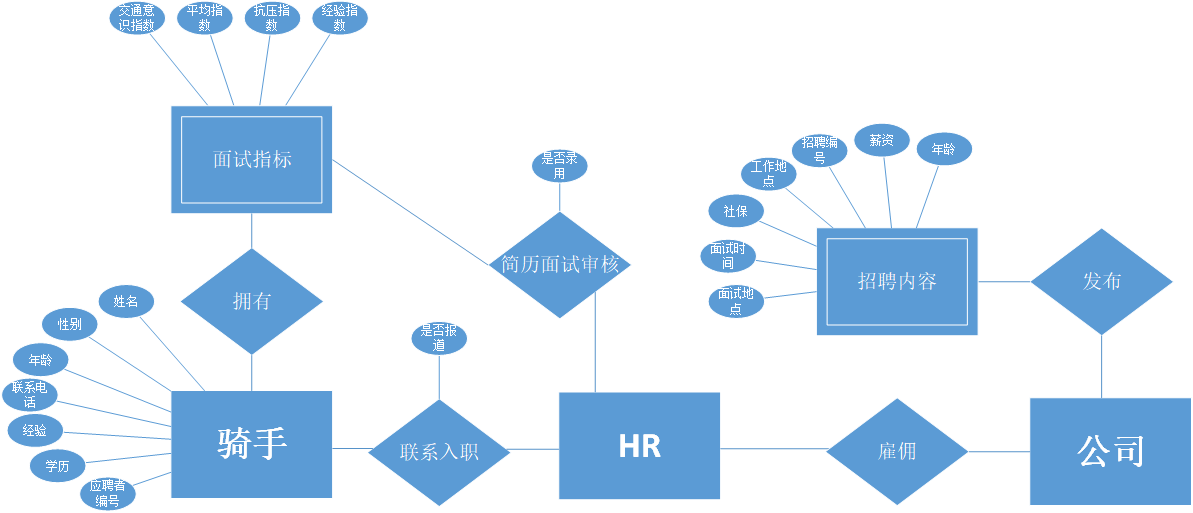
分类（Classification）——语义is member of

定义某一类概念作为现实世界中一组对象的类型，将一组具有某些共同特性和行为的对象抽象为一个实体，对象和实体之间是“is member of ”的关系

聚集（Aggregation）——语义is part of

定义某个类型的组成成分。将对象的类型的组成成分抽象为实体的属性。抽象了对象内部类型和成分的“is part of”的语义。

3.2局部E-R图的设计



（图3.1）

4个实体转化成对应的关系模式：  
骑手(应聘者编号，姓名，性别，年龄，联系电话，学历)

审核结果(应聘者编号，交通意识指数，经验指数，抗压指数，平均指数，是否录用)

招聘内容表(招聘编号，最高年龄，最低年龄，基础薪资，招聘起始日期，招聘截止日期，招聘面试地点)

报道结果(应聘者编号，联系电话，是否报道)

局部的E-R图中共包含4个联系：  
该E-R图由5个实体：骑手，HR，面试指标，招聘内容，公司组成，其中面试指标和招聘内容是弱实体集合

1)骑手有7个属性：应聘者编号（主），姓名，性别，年龄，联系电话，经验，学历

2)面试指标有5个属性：应聘者编号（主），抗压指标，经验指标，交通意识指标

3)招聘内容有7个属性：年龄，薪资，招聘编号（主），工作地点，面试时间，面试地点，社保

4)公司无属性

5)HR没有属性

该E-R图由5个关系组成：

1)骑手-拥有-面试指标 m:n

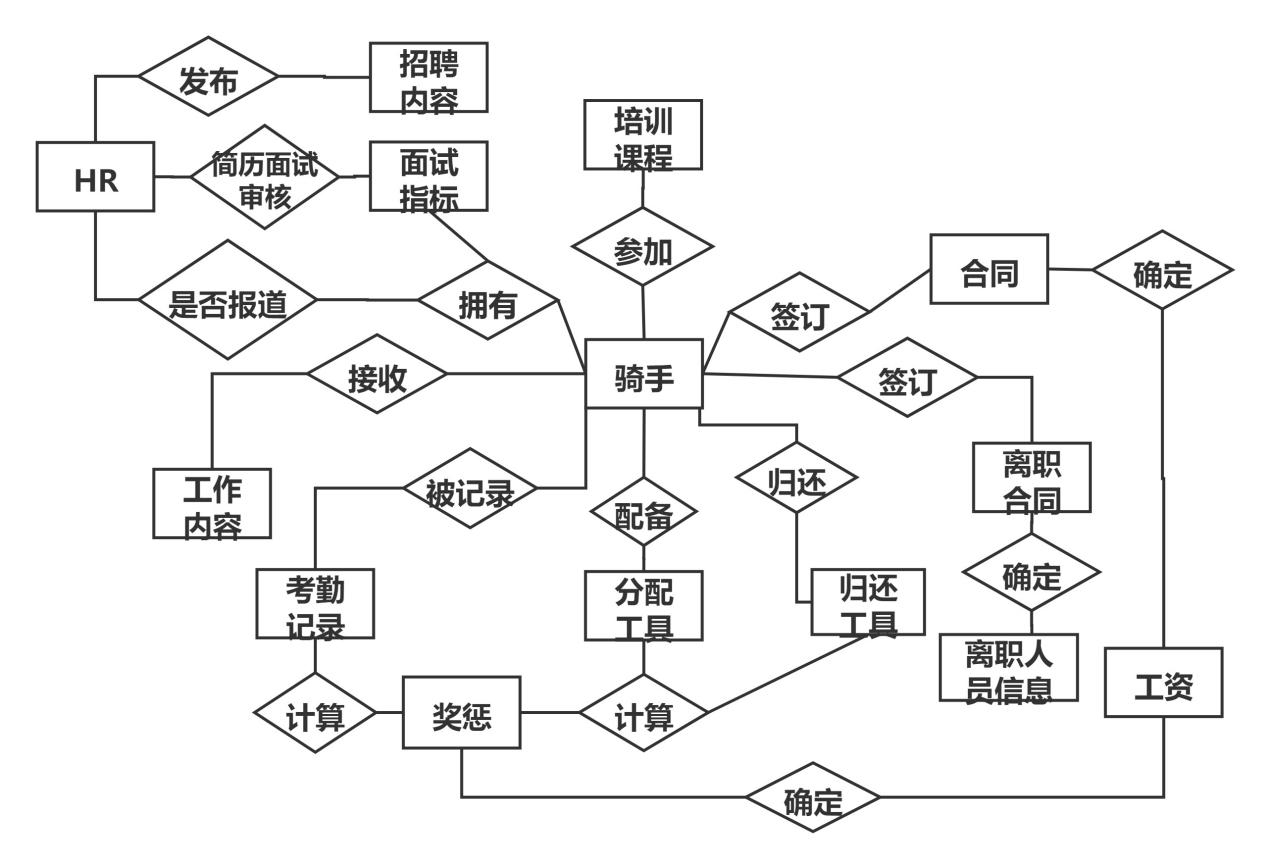
2)HR-简历面试审核-面试指标 审核关系中派生出是否录用属性 1:n

3)HR-联系入职-骑手 入职关系中派生出是否报道属性 1:n

4)公司-雇佣-HR 1:1(HR设是一个部门)

5)公司-发布-招聘内容 1：n

3.3总体E-R图合成

图（3.2）

四．逻辑模式设计

把概念结构设计阶段设计好的基本E-R图转换为与选用DBMS产品所支持的数据模型相符合的逻辑结构:

4.1数据逻辑结构

骑手：Rider

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 含义 | 类型 | 长度 | 备注 |
| cdanum | 应聘者编号 | numeric(6,0) | 5 | 主键，not  null ，位编号在numeric(6,0)范围内 |
| name | 姓名 | char(5) | 5 | Not null，五个汉字以内 |
| sex | 性别 | bool | 1 | Not null，男女 |
| age | 年龄 | tinyint | 2 | Not null  0-80 |
| edu | 学历 | char(2) | 2 | Not null，两个汉字 |
| tphone | 联系电话 | numeric(11,0) | 9 | Not null |

招聘内容RecruitContent

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 含义 | 类型 | 长度 | 备注 |
| recruitnum | 招聘编号 | numeric(6,0) | 5 | 主键，not null 6位编号在char3范围内 |
| lowage | 最低年龄 | tinyint | 1 | Not null，0-80在char1范围内 |
| highage | 最高年龄 | tinyint | 1 | Not null，0-80在char1范围内 |
| basepay | 薪资(底薪) | smallint | 2 | Not null，3000-10000在char(2)范围内 |
| Ivplace | 面试地点 | varchar（30） | 30 | Not null，长度不确定 |
| bgdate | 起始日期 | smalldatetime | 4 | Not null，长度不确定 |
| ddate | 截止日期 | smalldatetime | 4 | Null |

审核结果表:AuditResults

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 含义 | 类型 | 长度 | 备注 |
| cdanum | 应聘者编号 | Numeric(6,0) | 5 | 主键，not null  6位编号在char3范围内 |
| resisindex | 抗压指数 | tinyint | 1 | Not null  0-99在范围char1内 |
| experindex | 经验指数 | tinyint | 1 | Not null  0-99在范围char1内 |
| trafficawareIndex | 交通意识指数 | tinyint | 1 | Not null  0-99在范围char1内 |
| averindex | 平均指数 | tinyint | 1 | 计算列前三列的平均，not null |
| ishire | 是否录用 | bit | 1 | Not null 是否录用 |

报道结果：ArrivalResult

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 含义 | 类型 | 长度 | 备注 |
| cdanum | 应聘者编号 | Numeric(6,0) | 5 | 主键，not null  6位编号 在char3 范围内 |
| tphone | 联系电话 | Numeric(11.0) | 9 | Not null ，11位电话号码用char5 在范围内 |
| isarrival | 是否报道 | bit | 1 | Not null ，是否来报道，可以用bit存储 |

4.2 对于各个数据类型的解释

1. cdanum 是一个6位数的编号：采用numeric(6,0)类型定义数据，其中numeric(6,0)

所占内存大小表：

（图4.1）

类型的第一个参数是数据的位数，而第二位数是指小数的位数，同下使用的numeric（）类型，不可为空

2）name是一个5个汉字以内的字符串，其中一个汉字在SQLServer 中占一个char，所以设定为char（5），不可为空

3）sex用1/0代替‘男’、‘女’，这样能够缩小该数据项所占的空间，用bit表示，其中男用1表示，女用0表示，不可为空

4）age是一个100左右以内的数字，tinyint占一个字节从-128到+127正好能够胜任，不可为空

5)edu 是一个两位字符的字符串，例如：‘本科’，‘硕士’，‘博士’用char（2）不可为空

6)tphone 按照中国的普通用户的手机号码11位数字定义，可由numerci(11,0)定义，不可为空

7) recruitnum是一个6位数的编号：采用numeric(6,0)类型定义数据，其中numeric(6,0)

8) lowage 是一个100左右以内的数字，tinyint占一个字节从-128到+127正好能够胜任，不可为空

9）highage是一个100左右以内的数字，tinyint占一个字节从-128到+127正好能够胜任，不可为空

10)basepay 是一个10000以内的数字，可以采用smallint。smallint从 -2^15 (-32,768) 到 2^15 - 1 (32,767) 的整型数据。存储大小为 2 个字节。不可为空

11）Ivplace是一个较长的字符串而且不确定长度，由[varchar](60)定义，不可为空

12）bgdate 是一个日期变量，可以由smalldate 定义，从1900 年 1 月 1 日记录到 2079 年 6 月 6 日，精确度为1分钟，完全能够胜任工作，不可为空

13）ddate同上，补充，smalldate 占4个字节

14) resisindex是百分制的分数，采用tinyint见上文，可以为空，便于触发器插入数据，等待后续数据填入

15）experindex同上

16）trafficawareIndex同上

17）averindex 是前三个分数的平均数，与前三个数的类型一样

18）ishire 用于表示是否录用两种状况，由0/1表示，只占1bit

19）isarrival用于表示是否报道两种情况，由0/1表示，只占1bit

五．具体实现

5.1数据库及表的创建

一．创建数据库：

CREATE DATABASE Recruitment

ON(

NAME = N'Recruitment', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\Recruitment.mdf' ,

SIZE = 5MB , MAXSIZE = 200MB , FILEGROWTH = 10%)

LOG ON

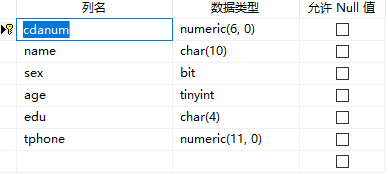
( NAME = N'Recruitment\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\ Recruitment.ndf' , SIZE = 2MB , MAXSIZE = 5MB , FILEGROWTH = 1MB )

GO

二．表的创建

TABLE\_1:rider 表

CREATE TABLE Rider(

 cdanum numeric(6,0) NOT NULL PRIMARY KEY,

name char(10) NOT NULL,

sex bit NOT NULL,

age tinyint NOT NULL,

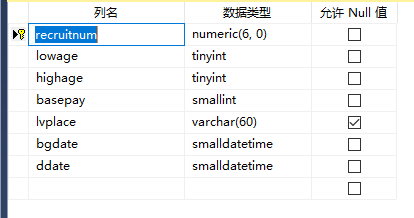
edu char(4) NOT NULL,

tphone numeric(11,0) NOT NULL, （图5.1）

CONSTRAINT age CHECK(age BETWEEN 0 AND 80)

)

TABLE\_2:

use AuditResults

CREATE TABLE RecruitContent(

recruitnum numeric(6,0) NOT NULL PRIMARY KEY,

lowage tinyint NOT NULL,

highage tinyint NOT NULL,

basepay smallint NOT NULL,

lvplace varchar(30) NOT NULL,

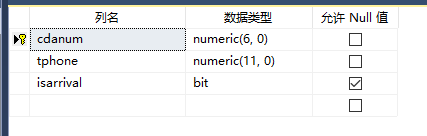
lvdate smalldate NOT NULL, （图5.2）

CONSTRAINT lowage CHECK(lowage BETWEEN 0 AND 80),

CONSTRAINT highage CHECK(highage BETWEEN 0 AND 80)

)

TABLE\_3:

use ArrivalResults

CREATE TABLE ArrivalResult

(

cdanum numeric(6,0) NOT NULL PRIMARY KEY,

tphone numeric(11,0) NOT NULL, （图5.3）

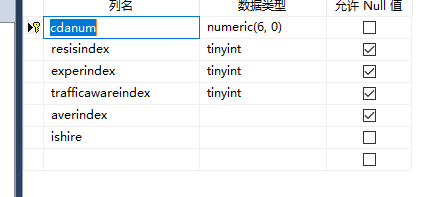
isarrival bit NULL

CONSTRAINT fr\_cda\_ar FOREIGN KEY (cdanum)references Rider(cdanum) ON DELETE CASCADE

)

TABLE\_4:

use Recruitment

create table AuditResults

(

cdanum numeric(6,0) NOT NULL PRIMARY KEY ,

resisindex tinyint NULL,

experindex tinyint NULL,

trafficawareindex tinyint NULL,

averindex as (resisindex+experindex+trafficawareindex)/3, （图5.4）

ishire as(case when (resisindex+experindex+trafficawareindex)/3> 60 then 1 else 0 end

CONSTRAINT fr\_cda\_adr FOREIGN KEY (cdanum)references Rider(cdanum) ON DELETE CASCADE)

5.2存储过程,约束,视图,索引设计

约束设计：

Rider:

CONSTRAINT age CHECK(age BETWEEN 0 AND 80)

RecruitContent：

CONSTRAINT lowage CHECK(lowage BETWEEN 0 AND 80),

CONSTRAINT highage CHECK(highage BETWEEN 0 AND 80)

约束设置年龄在0~80之间

ArrivalResults:

CONSTRAINT fr\_cda\_ar FOREIGN KEY (cdanum)references Rider(cdanum) ON DELETE CASCADE

设置cdanum为Rider（cdanum）的外键

Auditresults:

CONSTRAINT resisindex CHECK(resisindex BETWEEN 0 AND 99),

CONSTRAINT experindex CHECK(experindex BETWEEN 0 AND 99),

CONSTRAINT trafficawareindex CHECK(trafficawareindex BETWEEN 0 AND 99),

设置三项指标的范围为0~

averindex as (resisindex+experindex+trafficawareindex)/3

设置averindex自动为该三项指标的平均数

ishire as(case when (resisindex+experindex+trafficawareindex)/3> 60 then 1 else 0 end

设置如果平均数大于60自动录取该骑手

CONSTRAINT fr\_cda\_adr FOREIGN KEY (cdanum)references Rider(cdanum) ON DELETE CASCADE

设置cdanum为Rider（cdanum）的外键

触发器的设计：

1）当向rider插入数据时，向AuditResults传送数据，

create or alter trigger tri\_adr\_inser on Rider after insert--adr

as

if not exists(select cdanum from AuditResults where cdanum=(select cdanum from inserted))

begin

insert into AuditResults(cdanum)

values((select cdanum from inserted))

end

go

2）当在rider删除数据时，先在AuditResults删除对应的列

create or alter trigger tri\_adr\_dele on Rider for delete--ad

as

if exists(select cdanum from AuditResults where cdanum=(select cdanum from inserted))

begin

delete from AuditResults where cdanum=(select cdanum from deleted)

end

go

当向rider插入数据时，向Arrival传送数据，

create or alter trigger tri\_ad\_inser on Rider after insert

as

if not exists(select cdanum from ArrivalResult where cdanum=(select cdanum from inserted))

begin

insert AuditResult(cdanum,tphone)

values((select cdanum from inserted),(select tphone from inserted))

end

go

当在rider删除数据时，先在Arrival删除对应的列

create or alter trigger tri\_ad\_dele on Rider for delete

as

if exists(select cdanum from ArrivalResult where cdanum=(select cdanum from deleted))

begin

delete from ArrivalResult where cdanum=(select cdanum from inserted)

end

go

索引的设计：

Rider聚集索引：create index PK\_Recruit on Rider (cdanum)

AuditResults聚集索引:create index PK\_AuditRes on AuditResults(cdanum)

ArrivalResults聚集索引:create index PK\_Arrival on ArrivalResult(cdanum)

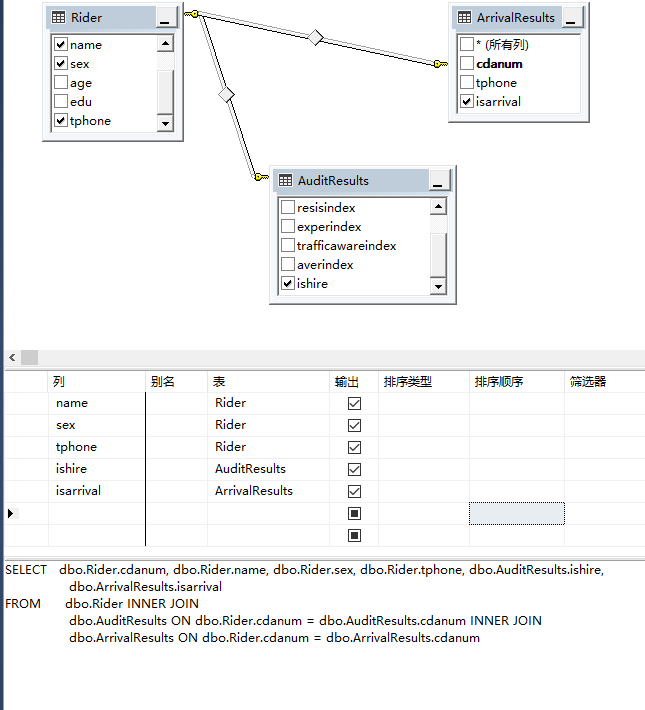
本次设计，基本上都是通过骑手编号来查询，因此，给三个表都上了 cdanum 的聚集索引大大减少查询时间

视图的设计：

HR\_View:

界面创建视图，其中汇聚三张表中最有效的数据，供HR部门高效地与骑手联系，

其中重要的信息为编号，姓名，性别，号码，是否录取，是否报道。

（图5.5）

代码为：

create view HR\_View

as

SELECT dbo.Rider.cdanum, dbo.Rider.name, dbo.Rider.sex, dbo.Rider.tphone, dbo.AuditResults.ishire,

dbo.ArrivalResults.isarrival

FROM dbo.Rider INNER JOIN

dbo.AuditResults ON dbo.Rider.cdanum = dbo.AuditResults.cdanum INNER JOIN

dbo.ArrivalResults ON dbo.Rider.cdanum = dbo.ArrivalResults.cdanum

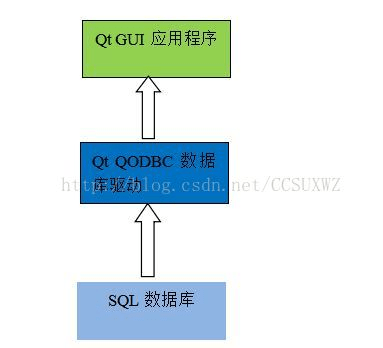
go

5.3 C++配置ODBC本地连接数据库

Qt连接SQL server数据库

由于课程设计需要，需要用qt设计一个界面来操作数据库，在建立数据库连接时，期间遇到各种问题。

Qt 连接SQL server数据大致可以概括为下图的三层模型，箭头代表他们之间的依赖关系  
（图5.6）



第一步：建立目标数据库  
在连接数据库之前，需要先在SQL Server中建立一个目标数据库，建立数据库和添加表的步骤这里不详述，

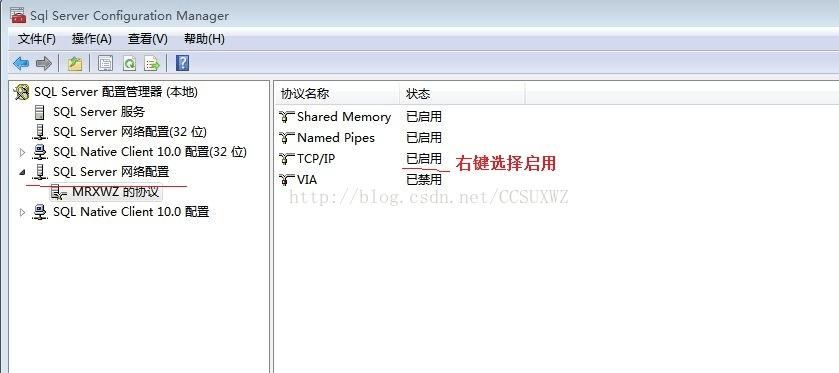
如建立一个名为Student 的数据库，其中有Student表、Course表和SC三个表。这里使用的是数据库版本为 SQL Server 2008 R2。

第二步：配置数据库数据源  
在配置数据源之前，要确保1433号端口已打开，1433端口，是SQL Server默认的端口，SQL Server服务使用两个端口：TCP-1433、UDP-1434。其中1433用于供SQL Server对外提供服务，1434用于向请求者返回SQL Server使用了哪个TCP/IP端口。

设置SQL Server 网络配置中的TCP/IP协议为已启用状态，并且检查其默认端口是否为1433端口。

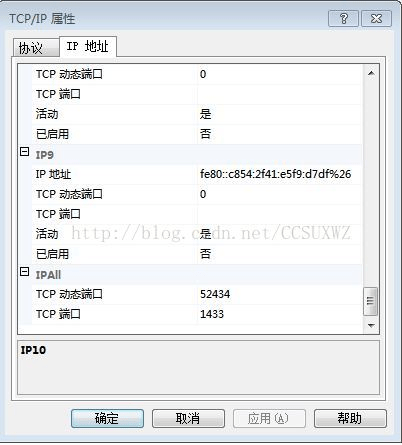
在开始菜单搜索SQL Server 配置管理器，选择SQL Server网络配置。

图（5.7）



右键TCP/IP，选择属性查看其默认端口，设置其为1433

（图5.8）



服务和端口设置好后接下来配置ODBC数据源。开放数据库连接（Open Database Connectivity，ODBC）是微软公司开放服务结构（WOSA，Windows Open Services Architecture）中有关数据库的一个组成部分，它建立了一组规范，并提供了一组对数据库访问的标准API（应用程序编程接口）

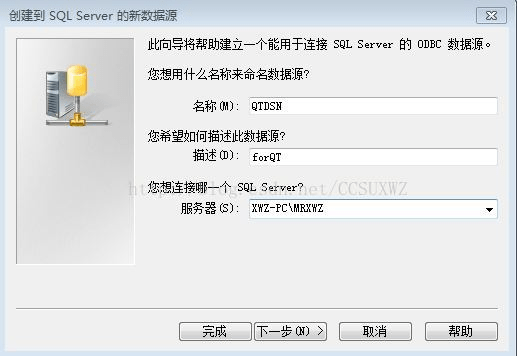
Qt数据库驱动并不能直接连接到SQL Server中的数据库，而是要通过配置ODBC数据源来进行连接。

开始->搜索ODBC->点击dBASE Files 选择添加

选择SQL Server

命名数据源名称并选择服务器，这里的服务器名即SQL Server 登录框中的服务器名，把它复制过来

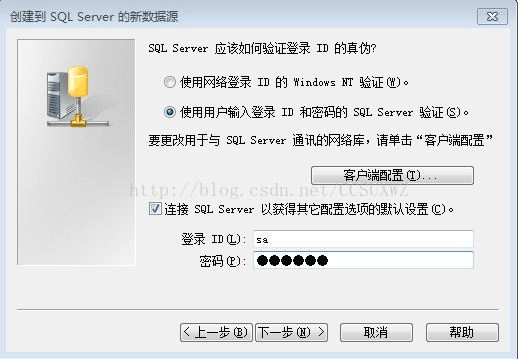
（图5.9）



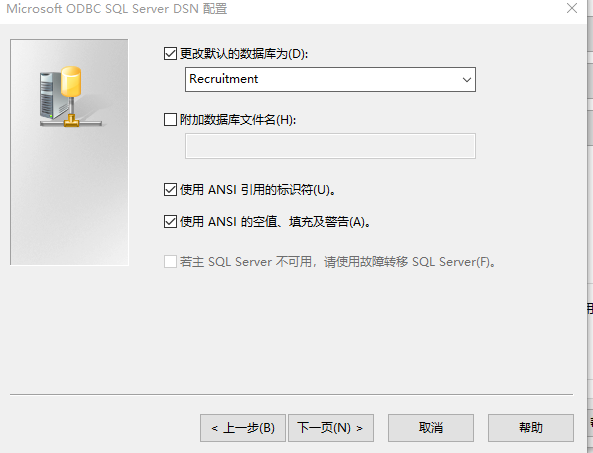
选择登录方式为SQL Sever验证方式，并设置登录名和密码，这里选择sa（超级用户）

注意：登录名必须是数据库中已存在的登录用户

（图5.10）

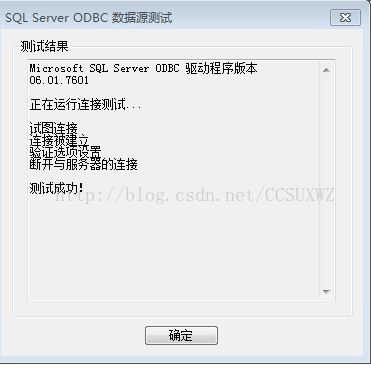


选择需要连接的数据库作为默认数据库

（图5.11）

选择下一步 ，测试数据源，成功后单击确定。

（图5.12）



第三步 Qt连接数据源  
新建一个Qt GUI工程，在其pro文件中加入QT += sql，设计界面并设计界面

5.4界面创建

5.4.1登录界面设计

//限于篇幅，代码内容

数据库连接代码：

//创建数据库对象选用ODBC作为数据源

db=QSqlDatabase::addDatabase("QODBC");

//数据源名为SQL SERVER

db.setDatabaseName(QString("DRIVER={SQL SERVER};"

"SERVER=%1;" //servername

"DATABASE=%2;" //sqlname

"UID=%3;" //loadid

"PWD=%4;" //password

).arg("LAPTOP-FFBS4IFA")

.arg("Recuritment")

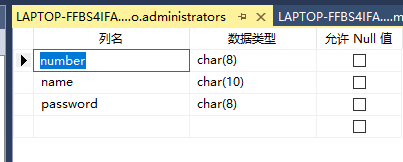
.arg("sa")

.arg("20203737")

);

//服务器名，数据库名，sqlserver账户，密码

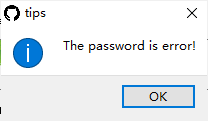
在后台连接数据库之前，需要创建一张表administrators，设置管理人员权限：

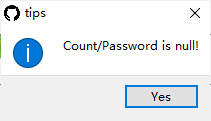
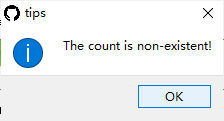
（图5.13）

其中，number是编号，name是账号，password是密码

通过连接到数据库后在Recruitment数据库中访问表administrators，查询是否存在账号，密码，以此完成登录功能。（图5.14）

此时有三种情况：

1）账号存在，密码错误，输出（图5.15）

2）账号密码都不存在，输出（图5.16）

3）账号密码若有空，输出

（图5.17）

代码一览：

if(ui->MassgeradioBtn->isChecked())//当点击‘登录’按钮

{

query.exec(QString("select name from administrators where name='%1'").arg(CountStr));//向后台发布查询语句

query.next();

index = query.value(0).toString().simplified();

//得到数据库表的第一条信息

if(index != CountStr)//如果没有这个管理员账号

{

QMessageBox::information(this,"tips","The count is non-existent!");//提示 这个账号不存在

qDebug()<<CountStr<<index;

return;

}

query.exec(QString("select password from administrators where password='%1'").arg(PassWordStr));

query.next();

password = query.value(0).toString().simplified();

if(password == PassWordStr)//密码输入正确

{

qDebug()<<password<<PassWordStr;

this->close();

HomePage \* showWidget = new HomePage;

//创建home功能界面

showWidget->show();

}

转而得到如下homepage：

（图5.18）

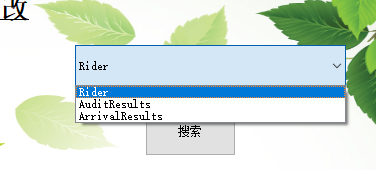
5.4.2查询修改界面设计

点击修改骑手信息，进入查询修改界面：

#### 1）查询功能：

（图5.19）

可以通过右上角的表combobox选择表的查询的表（不可为空），再通过左上角的输入框，输入需要查询骑手的个人信息，如果为空，则查询整张表的数据。

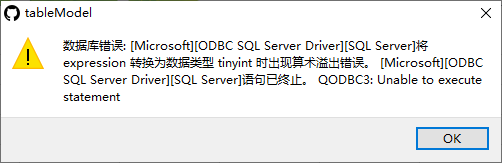
（图5.20）

#### 2）修改功能：

再者可以对每个单元格直接双击修改内容，最终点击修改按钮后，将数据传送到数据库，迅速完成修改。

（图5.21）

如果修改数据不符合约束，或者和数据类型不匹配，发出由数据库传来的报错：

（图5.22）

5.4.3添加骑手界面设计

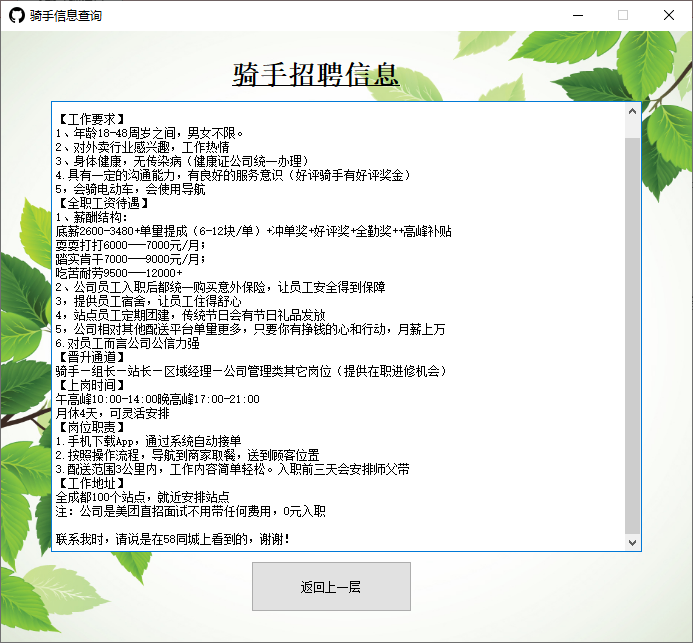
点击添加应聘骑手：

（图5.23）

通过输入各个数据后，点击确定 后，会将数据传送到数据库后台，用sql语句增加该人物，插入成功后激发触发器，向其他两个AuditResults和ArrivalResults表中插入数据，没有包含的项，用null代替，之后再用查询修改界面，完善数据。

如果输入数据不符合约束，或者和数据类型不匹配，发出insert failed指令。

5.4.4骑手招聘信息界面设计

（图5.24）

这个功能就是为了返回一个长长的字符串，其中将RecruitContent表中的数据串联到该字符中，完成这一功能的实现，该表内容不添加修改操作。

5.4.5骑手招聘信息界面设

点击‘删除骑手’按钮，进入：

（图5.25）

通过编号查找出骑手后，确认骑手信息后，点击删除骑手，此时激发删除触发器，对AuditResults表和ArrivalResutls表删除对应cdanum的数据行，从三张表进行删除操作。