

**Database design document**

**数据库设计文档**

人才市场大数据智能分析展示系统

2019

**目录**

**[第一章 前言 3](#_Toc1654619296)**

[1.1编写目的 3](#_Toc1420762869)

[1.2数据库环境说明 3](#_Toc890868290)

[1.3数据库命名规则 3](#_Toc567363146)

[1.4参考资料 3](#_Toc845002142)

**[第二章 数据结构设计 3](#_Toc641642983)**

[2.1 总体设计 3](#_Toc1578223694)

[2.2表的清单 5](#_Toc1635100961)

[2.3引用的清单 5](#_Toc1941104515)

[2.4 表的数据结构 6](#_Toc1719502028)

[2.5表的创建代码 11](#_Toc983146917)

# 第一章 前言

## 1.1编写目的

本文档是人才市场大数据智能分析展示系统概要设计和详细设计文档的组成部分，编写数据库设计文档的目的是：明确数据库的表名、字段名等数据信息，用来指导后期的数据库脚本的开发，本文档遵循《SQL数据库设计和开发规范》。本文档的读者对象是需求人员、系统设计人员、开发人员、测试人员。

## 1.2数据库环境说明

* 数据库：MySql 5.7
* 数据库建模工具：PowerDesigner 16.5
* 数据库管理工具：Navicat Premium 12

## 1.3数据库命名规则

数据库中表及字段等对象的命名都以统一的规则，采用大小写敏感的形式，各种对象命名长度不要超过 30 个字符，这样便于应用系统适应不同的数据库平台。对于本系统，数据库名为fuchuang，表名以英文（表的用途）+数字，字段名以”Data\_”加对应属性英文名开头。

## 1.4参考资料

* 《SQL数据库设计和开发规范》
* 《软件设计文档国家标准数据库设计文档GB8567——88》

# 第二章 数据结构设计

## 2.1 总体设计

人才市场大数据智能分析展示系统的数据库设计基于对被系统业务的了解，在遵守数据库第 Ｈ范式的设计原则基础上，来进行系统数据库的设计。

在数据库设计进行前，人才市场大数据智能分析展示系统确定了以下两个设计的基本原则。（1）数据库表结构设计基于第Ｈ范式，但是特殊情况下可违反第三范式。在知识库系统的数据 库表结构设计上，争取按照第三范式的要求，不对数据进行冗余的处理，但是在实际的情况中，有些功能如果不进行冗余处理，可能会涉及到多个表之间的关联查询，送种查 询在效率上较低，影响到用户的使用，而如果在这些功能中使用冗余的方式，则会有效的提高查询的效率，在这种情况下，建议系统的开发不能完全遵守第三范式。（2）系统索引的设置基于按照实际的sql语句进行，尽量使用b-tree索引。对于数据库设计来说，索引的设计是尤其重要的一个环节，索引分为很多类型，联合索引，位图索引等各种索引的使用方式以及含义都不尽相同，在此情况下，建议使用 b-tree 索引，因为 b-tree 索引具有良好的定位特性，其常被用于对检索时间要求苛刻的场合，与内存相比，硬盘必须花成倍的时间来存取一个数据元素，这是因为硬盘的机械部件读写数据的速度远远赶 不上纯电子媒体的内存。与一个结点两个分支的二元树相比，B-tree 利用多个分支（称 为子树）的结点，减少获取记录时所经历的结点数，从而达到节省存取时间的目的。

参考《需求规格说明书》，并通过综合分析，可得出系统的 ER 图，如图 1.1 所示。



图 2-1 系统的ER图

## 2.2表的清单

|  |  |
| --- | --- |
| 名称 | 代码 |
| 2019年企业各招聘渠道比例预测表 | info11 |
| 2019年各学历层次工资预测表 | info12 |
| 2019年热门城市就业推荐指数表 | info9 |
| 2019年行业信息预测表 | info10 |
| 各学历人数比例表 | info2 |
| 学历与工资关系表 | info4 |
| 实时招聘信息表 | empty\_data |
| 总招聘信息表 | main\_data |
| 总招聘信息表2 | main\_data2 |
| 总招聘信息表3 | main\_data3 |
| 招聘信息数据量表 | con |
| 热门城市工资情况表 | info6 |
| 热门城市招聘数据表 | info7 |
| 热门城市招聘需求表 | info5 |
| 热门行业待遇等级表 | info3 |
| 热门行业招聘需求表 | info1 |
| 部分城市工作满意度表 | info8 |
| 预测招聘信息表 | recommend |

## 

## 2.3引用的清单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 父表格 | 子表格 | 外键列 |
| Reference\_1 | Reference\_1 | 总招聘信息表 | 热门行业招聘需求表 | 行业 |
| Reference\_3 | Reference\_3 | 总招聘信息表 | 2019年行业信息预测表 | 行业 |
| Reference\_4 | Reference\_4 | 总招聘信息表2 | 热门城市工资情况表 | 城市 |
| Reference\_5 | Reference\_5 | 总招聘信息表2 | 热门城市招聘需求表 | 城市 |
| Reference\_6 | Reference\_6 | 总招聘信息表2 | 热门城市招聘数据表 | 城市 |
| Reference\_7 | Reference\_7 | 总招聘信息表2 | 2019年热门城市就业推荐指数表 | 城市 |
| Reference\_8 | Reference\_8 | 总招聘信息表2 | 部分城市工作满意度表 | 城市 |
| Reference\_9 | Reference\_9 | 总招聘信息表 | 热门行业待遇等级表 | 行业 |
| Reference\_10 | Reference\_10 | 总招聘信息表3 | 2019年各学历层次工资预测表 | 学历 |
| Reference\_11 | Reference\_11 | 总招聘信息表3 | 各学历人数比例表 | 学历 |
| Reference\_12 | Reference\_12 | 总招聘信息表3 | 学历与工资关系表 | 学历2 |
| Reference\_13 | Reference\_13 | 实时招聘信息表 | 预测招聘信息表 | 城市 |

## 2.4 表的数据结构

数据库中共有 18 张表，这些表的详细数据结构请参考表 2.1～2.18。首先，表 2.1 为总招聘信息表，表名为 main\_data，存储了收集的所有招聘数据信息，具体数据结构如下：

表2-1 总招聘信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  |  |
| 行业 | Data\_indus | varchar | 100 | pk |
| 职业 | Data\_job | varchar | 100 |  |
| 城市 | Data\_city | varchar | 100 |  |
| 工资 | Data\_salary | double |  |  |
| 工作经验 | Data\_exper | varchar | 100 |  |
| 公司类型 | Data\_comtype | varchar | 100 |  |
| 公司 | Data\_com | varchar | 100 |  |
| 学历 | Data\_edu | varchar | 100 |  |

表 2.2 为热门行业招聘需求表，表名为 info1，存储了人们行业招聘需求的数据信息，具体数据结构如下：

表2-2 热门行业招聘需求表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| 行业 | Data\_indus | varchar | 100 | pk,fk |
| 人数 | number | int |  |  |

表 2.3 为各学历人数比例表，表名为 info2，存储了各学历人数比例的数据信息，具体数据结构如下：

表2-3 各学历人数比例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  | pk |
| 人数 | number | int |  |  |
| 学历 | Data\_edu | varchar | 100 | fk |

表 2.4 为热门行业待遇等级表，表名为 info3，存储了热门行业待遇等级的数据信息，具体数据结构如下：

表2-4 热门行业待遇等级表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  | pk |
| 行业 | Data\_indus | varchar | 100 | fk |
| 最低工资 | min | double |  |  |
| 最高工资 | max | double |  |  |
| 平均工资 | avg | double |  |  |
| 待遇等级 | level | double |  |  |

表 2.5 为学历与工资关系表，表名为 info4，存储了学历与工资对应关系的数据信息，具体数据结构如下：

表2-5 学历与工资关系表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  | pk |
| 学历 | Data\_edu | varchar | 100 | fk |
| 最高工资 | max | double |  |  |
| 最低工资 | min | double |  |  |
| 行业 | Data\_indus | varchar | 100 |  |

表 2.6 为热门城市招聘需求表，表名为 info5，存储了热门城市招聘需求的数据信息，具体数据结构如下：

表2-6 热门城市招聘需求表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  | pk |
| 城市 | Data\_city | varchar | 100 | fk |
| 人数 | value | int |  |  |

表 2.7 为热门城市工资情况表，表名为 info6，存储了热门城市工作情况的数据信息，具体数据结构如下：

表2-7 热门城市工资情况表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  | pk |
| 城市 | Data\_city | varchar | 100 | fk |
| 人数 | con | int |  |  |
| 最高工资 | max | double |  |  |
| 平均工资 | avg | double |  |  |
| 最低工资 | min | double |  |  |

表 2.8 为热门城市招聘数据表，表名为 info7，存储了热门城市招聘的数据信息，具体数据结构如下：

表2-8 热门城市招聘数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  | pk |
| 城市 | Data\_city | varchar | 100 | fk |
| 人数 | con | int |  |  |
| 标准差 | std | double |  |  |
| 平均工资 | avg | double |  |  |

表 2.9 为部分城市工作满意度表，表名为 info8，存储了中国部分城市工作满意度的数据信息，具体数据结构如下：

表2-9 部分城市工作满意度表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  | pk |
| 满意度 | level | double |  |  |
| 城市 | Data\_city | varchar | 100 | fk |

表 2.10 为2019年热门城市就业推荐指数表，表名为 info9，存储了预测得出的2019年热门城市就业推荐指数的数据信息，具体数据结构如下：

表2-10 2019年热门城市就业推荐指数表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  | pk |
| 城市 | Data\_city | varchar | 100 | fk |
| 工资 | Data\_salary | double |  |  |
| 推荐指数 | level | double |  |  |
| 人数 | con | int |  |  |

表 2.11 为2019年行业信息预测表，表名为 info10，存储了所预测的2019行业的数据信息，具体数据结构如下：

表2-11 2019年行业信息预测表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  | pk |
| 行业 | Data\_indus | varchar | 100 | fk |
| 人数 | con | int |  |  |
| 工资 | Data\_salary | double |  |  |
| 推荐等级 | level | int |  |  |

表 2.12 为2019年企业各招聘渠道比例预测表，表名为 info11，存储了2019年企业各招聘渠道的比例信息，具体数据结构如下：

表2-12 2019年企业各招聘渠道比例预测表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  | pk |
| 社招人数 | citynum | int |  |  |
| 校招人数 | total | int |  |  |

表 2.13 为2019年各学历层次工资预测表，表名为 info12，存储了所预测的2019年各学历层次工资的数据信息，具体数据结构如下：

表2-13 2019年各学历层次工资预测表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  | pk |
| 人数 | con | int |  |  |
| 学历 | Data\_edu | varchar | 100 | fk |
| 平均工资 | avgsal | double |  |  |
| 最高工资 | maxsal | double |  |  |
| 最低工资 | minsal | double |  |  |

表 2.14 为实时招聘信息表，表名为 empty\_data，存储了实时获取的招聘数据信息，具体数据结构如下：

表2-14 实时招聘信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  |  |
| 职业 | Data\_job | varchar | 100 | pk |
| 城市 | Data\_city | varchar | 100 |  |
| 工资 | Data\_salary | double |  |  |
| 工作经验 | Data\_exper | varchar | 100 |  |
| 学历 | Data\_edu | varchar | 100 |  |
| 公司类型 | Data\_comtype | varchar | 100 |  |
| 公司 | Data\_com | varchar | 100 |  |
| 行业 | Data\_indus | varchar | 100 |  |

表 2.15 为招聘信息数据量表，表名为 con，存储了所收集的招聘信息的条数，具体数据结构如下：

表2-15 招聘信息数据量表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| 数据量 | con | int |  |  |

表 2.16 为总招聘信息表2，表名为 main\_data2，存储了总的招聘数据信息，具体数据结构如下：

表2-16 总招聘信息表2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  |  |
| 行业 | Data\_indus | varchar | 100 | pk |
| 职业 | Data\_job | varchar | 100 |  |
| 城市 | Data\_city | varchar | 100 |  |
| 工资 | Data\_salary | double |  |  |
| 工作经验 | Data\_exper | varchar | 100 |  |
| 公司类型 | Data\_comtype | varchar | 100 |  |
| 公司 | Data\_com | varchar | 100 |  |
| 学历 | Data\_edu | varchar | 100 |  |

表 2.17 为总招聘信息表3，表名为 main\_data3，存储了总的招聘数据信息，具体数据结构如下：

表2-17 总招聘信息表3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  |  |
| 行业 | Data\_indus | varchar | 100 | pk |
| 职业 | Data\_job | varchar | 100 |  |
| 城市 | Data\_city | varchar | 100 |  |
| 工资 | Data\_salary | double |  |  |
| 工作经验 | Data\_exper | varchar | 100 |  |
| 公司类型 | Data\_comtype | varchar | 100 |  |
| 公司 | Data\_com | varchar | 100 |  |
| 学历 | Data\_edu | varchar | 100 |  |

表 2.18 为预测招聘信息表，表名为 recommend，存储了所预测的招聘信息，具体数据结构如下：

表2-18 预测招聘信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 代码 | 数据类型 | 长度 | 备注 |
| ID | Data\_id | int |  | pk |
| 工作 | Data\_job | varchar | 100 |  |
| 城市 | Data\_city | varchar | 100 | fk |
| 平均工资 | avgsal | double |  |  |

## 2.5表的创建代码

if exists(select 1 from sys.sysforeignkey where role='FK\_INFO1\_REFERENCE\_MAIN\_DAT') then

alter table info1

delete foreign key FK\_INFO1\_REFERENCE\_MAIN\_DAT

end if;

if exists(select 1 from sys.sysforeignkey where role='FK\_INFO10\_REFERENCE\_MAIN\_DAT') then

alter table info10

delete foreign key FK\_INFO10\_REFERENCE\_MAIN\_DAT

end if;

if exists(select 1 from sys.sysforeignkey where role='FK\_INFO12\_REFERENCE\_MAIN\_DAT') then

alter table info12

delete foreign key FK\_INFO12\_REFERENCE\_MAIN\_DAT

end if;

if exists(select 1 from sys.sysforeignkey where role='FK\_INFO2\_REFERENCE\_MAIN\_DAT') then

alter table info2

delete foreign key FK\_INFO2\_REFERENCE\_MAIN\_DAT

end if;

if exists(select 1 from sys.sysforeignkey where role='FK\_INFO3\_REFERENCE\_MAIN\_DAT') then

alter table info3

delete foreign key FK\_INFO3\_REFERENCE\_MAIN\_DAT

end if;

if exists(select 1 from sys.sysforeignkey where role='FK\_INFO4\_REFERENCE\_MAIN\_DAT') then

alter table info4

delete foreign key FK\_INFO4\_REFERENCE\_MAIN\_DAT

end if;

if exists(select 1 from sys.sysforeignkey where role='FK\_INFO5\_REFERENCE\_MAIN\_DAT') then

alter table info5

delete foreign key FK\_INFO5\_REFERENCE\_MAIN\_DAT

end if;

if exists(select 1 from sys.sysforeignkey where role='FK\_INFO6\_REFERENCE\_MAIN\_DAT') then

alter table info6

delete foreign key FK\_INFO6\_REFERENCE\_MAIN\_DAT

end if;

if exists(select 1 from sys.sysforeignkey where role='FK\_INFO7\_REFERENCE\_MAIN\_DAT') then

alter table info7

delete foreign key FK\_INFO7\_REFERENCE\_MAIN\_DAT

end if;

if exists(select 1 from sys.sysforeignkey where role='FK\_INFO8\_REFERENCE\_MAIN\_DAT') then

alter table info8

delete foreign key FK\_INFO8\_REFERENCE\_MAIN\_DAT

end if;

if exists(select 1 from sys.sysforeignkey where role='FK\_INFO9\_REFERENCE\_MAIN\_DAT') then

alter table info9

delete foreign key FK\_INFO9\_REFERENCE\_MAIN\_DAT

end if;

if exists(select 1 from sys.sysforeignkey where role='FK\_RECOMMEN\_REFERENCE\_EMPTY\_DA') then

alter table recommend

delete foreign key FK\_RECOMMEN\_REFERENCE\_EMPTY\_DA

end if;

drop table if exists con;

drop table if exists empty\_data;

drop table if exists info1;

drop table if exists info10;

drop table if exists info11;

drop table if exists info12;

drop table if exists info2;

drop table if exists info3;

drop table if exists info4;

drop table if exists info5;

drop table if exists info6;

drop table if exists info7;

drop table if exists info8;

drop table if exists info9;

drop table if exists main\_data;

drop table if exists main\_data2;

drop table if exists main\_data3;

drop table if exists recommend;

/\*==============================================================\*/

/\* Table: con \*/

/\*==============================================================\*/

create table con

(

con int not null,

constraint PK\_CON primary key clustered (con)

);

/\*==============================================================\*/

/\* Table: empty\_data \*/

/\*==============================================================\*/

create table empty\_data

(

Data\_id varchar(100) not null,

Data\_job varchar(100) not null,

Data\_city varchar(100) not null,

Data\_salary double not null,

Data\_exper varchar(100) not null,

Data\_edu varchar(100) not null,

Data\_comtype varchar(100) not null,

Data\_com varchar(100) not null,

Data\_indus varchar(100) not null,

constraint PK\_EMPTY\_DATA primary key clustered (Data\_city)

);

/\*==============================================================\*/

/\* Table: info1 \*/

/\*==============================================================\*/

create table info1

(

Data\_indus varchar(100) not null,

number double not null,

constraint PK\_INFO1 primary key clustered (Data\_indus)

);

/\*==============================================================\*/

/\* Table: info10 \*/

/\*==============================================================\*/

create table info10

(

Data\_id int not null,

Data\_indus varchar(100) null,

con int not null,

Data\_salary double not null,

level int not null,

constraint PK\_INFO10 primary key clustered (Data\_id)

);

/\*==============================================================\*/

/\* Table: info11 \*/

/\*==============================================================\*/

create table info11

(

Data\_id int not null,

citynum double not null,

total double not null,

constraint PK\_INFO11 primary key clustered (Data\_id)

);

/\*==============================================================\*/

/\* Table: info12 \*/

/\*==============================================================\*/

create table info12

(

Data\_id int not null,

con int not null,

Data\_edu varchar(100) not null,

avgsal double not null,

maxsal double not null,

minsal double not null,

constraint PK\_INFO12 primary key clustered (Data\_id)

);

/\*==============================================================\*/

/\* Table: info2 \*/

/\*==============================================================\*/

create table info2

(

Data\_id int not null,

number int not null,

Data\_edu varchar(100) not null,

constraint PK\_INFO2 primary key clustered (Data\_id)

);

/\*==============================================================\*/

/\* Table: info3 \*/

/\*==============================================================\*/

create table info3

(

Data\_id int not null,

Data\_indus varchar(100) not null,

min double not null,

max double not null,

avg double not null,

level double not null,

constraint PK\_INFO3 primary key clustered (Data\_id)

);

/\*==============================================================\*/

/\* Table: info4 \*/

/\*==============================================================\*/

create table info4

(

Data\_id int not null,

Data\_edu varchar(100) null,

max double not null,

min double not null,

Data\_indus varchar(100) null,

constraint PK\_INFO4 primary key clustered (Data\_id)

);

/\*==============================================================\*/

/\* Table: info5 \*/

/\*==============================================================\*/

create table info5

(

Data\_id int not null,

Data\_city varchar(100) not null,

value int not null,

constraint PK\_INFO5 primary key clustered (Data\_id)

);

/\*==============================================================\*/

/\* Table: info6 \*/

/\*==============================================================\*/

create table info6

(

Data\_id int not null,

Data\_city varchar(100) not null,

con int not null,

max double not null,

avg double not null,

min double not null,

constraint PK\_INFO6 primary key clustered (Data\_id)

);

/\*==============================================================\*/

/\* Table: info7 \*/

/\*==============================================================\*/

create table info7

(

Data\_id int not null,

Data\_city varchar(100) not null,

con int not null,

std int not null,

avg double not null,

constraint PK\_INFO7 primary key clustered (Data\_id)

);

/\*==============================================================\*/

/\* Table: info8 \*/

/\*==============================================================\*/

create table info8

(

Data\_id int not null,

level double not null,

Data\_city varchar(100) null,

constraint PK\_INFO8 primary key clustered (Data\_id)

);

/\*==============================================================\*/

/\* Table: info9 \*/

/\*==============================================================\*/

create table info9

(

Data\_id int not null,

Data\_city varchar(100) null,

Data\_salary double not null,

level double not null,

con int not null,

constraint PK\_INFO9 primary key clustered (Data\_id)

);

/\*==============================================================\*/

/\* Table: main\_data \*/

/\*==============================================================\*/

create table main\_data

(

Data\_id varchar(100) not null,

Data\_indus varchar(100) not null,

Data\_job varchar(100) not null,

Data\_city varchar(100) not null,

Data\_salary double not null,

Data\_exper varchar(100) not null,

Data\_comtype varchar(100) not null,

Data\_com varchar(100) not null,

Data\_edu varchar(100) not null,

constraint PK\_MAIN\_DATA primary key clustered (Data\_indus)

);

/\*==============================================================\*/

/\* Table: main\_data2 \*/

/\*==============================================================\*/

create table main\_data2

(

Data\_id varchar(100) not null,

Data\_indus varchar(100) not null,

Data\_job varchar(100) not null,

Data\_city varchar(100) not null,

Data\_salary double not null,

Data\_exper varchar(100) not null,

Data\_comtype varchar(100) not null,

Data\_com varchar(100) not null,

Data\_edu varchar(100) not null,

constraint PK\_MAIN\_DATA2 primary key clustered (Data\_city)

);

/\*==============================================================\*/

/\* Table: main\_data3 \*/

/\*==============================================================\*/

create table main\_data3

(

Data\_id varchar(100) not null,

Data\_indus varchar(100) not null,

Data\_job varchar(100) not null,

Data\_city varchar(100) not null,

Data\_salary double not null,

Data\_exper varchar(100) not null,

Data\_comtype varchar(100) not null,

Data\_com varchar(100) not null,

Data\_edu varchar(100) not null,

constraint PK\_MAIN\_DATA3 primary key clustered (Data\_edu)

);

/\*==============================================================\*/

/\* Table: recommend \*/

/\*==============================================================\*/

create table recommend

(

Data\_id int not null,

Data\_job varchar(100) not null,

Data\_city varchar(100) not null,

avgsal double not null,

constraint PK\_RECOMMEND primary key clustered (Data\_id)

);

alter table info1

add constraint FK\_INFO1\_REFERENCE\_MAIN\_DAT foreign key (Data\_indus)

references main\_data (Data\_indus)

on update restrict

on delete restrict;

alter table info10

add constraint FK\_INFO10\_REFERENCE\_MAIN\_DAT foreign key (Data\_indus)

references main\_data (Data\_indus)

on update restrict

on delete restrict;

alter table info12

add constraint FK\_INFO12\_REFERENCE\_MAIN\_DAT foreign key (Data\_edu)

references main\_data3 (Data\_edu)

on update restrict

on delete restrict;

alter table info2

add constraint FK\_INFO2\_REFERENCE\_MAIN\_DAT foreign key (Data\_edu)

references main\_data3 (Data\_edu)

on update restrict

on delete restrict;

alter table info3

add constraint FK\_INFO3\_REFERENCE\_MAIN\_DAT foreign key (Data\_indus)

references main\_data (Data\_indus)

on update restrict

on delete restrict;

alter table info4

add constraint FK\_INFO4\_REFERENCE\_MAIN\_DAT foreign key (Data\_edu)

references main\_data3 (Data\_edu)

on update restrict

on delete restrict;

alter table info5

add constraint FK\_INFO5\_REFERENCE\_MAIN\_DAT foreign key (Data\_city)

references main\_data2 (Data\_city)

on update restrict

on delete restrict;

alter table info6

add constraint FK\_INFO6\_REFERENCE\_MAIN\_DAT foreign key (Data\_city)

references main\_data2 (Data\_city)

on update restrict

on delete restrict;

alter table info7

add constraint FK\_INFO7\_REFERENCE\_MAIN\_DAT foreign key (Data\_city)

references main\_data2 (Data\_city)

on update restrict

on delete restrict;

alter table info8

add constraint FK\_INFO8\_REFERENCE\_MAIN\_DAT foreign key (Data\_city)

references main\_data2 (Data\_city)

on update restrict

on delete restrict;

alter table info9

add constraint FK\_INFO9\_REFERENCE\_MAIN\_DAT foreign key (Data\_city)

references main\_data2 (Data\_city)

on update restrict

on delete restrict;

alter table recommend

add constraint FK\_RECOMMEN\_REFERENCE\_EMPTY\_DA foreign key (Data\_city)

references empty\_data (Data\_city)

on update restrict

on delete restrict;