|  |  |
| --- | --- |
| 编号 |  |

《数据库原理》数据库设计报告

\*\*\*\*\*\*数据库设计

学号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_姓名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_姓名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_姓名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_姓名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_姓名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2022级\*\*\*\*\*\*\*\*班

2022.12

目 录

[1 项目背景 1](#_Toc37864946)

[2 需求分析 2](#_Toc37864947)

[2.1 信息要求 2](#_Toc37864948)

[2.2 处理要求 2](#_Toc37864949)

[2.3 安全性和完整性要求 2](#_Toc37864950)

[3 概念结构设计 3](#_Toc37864951)

[3.1 确定实体 3](#_Toc37864952)

[3.2 局部E-R图设计 3](#_Toc37864953)

[3.3 全局E-R图设计 3](#_Toc37864954)

[4 逻辑结构设计 4](#_Toc37864955)

[4.1 关系模式设计 4](#_Toc37864956)

[4.2 关系模式优化 4](#_Toc37864957)

[5 物理结构设计 5](#_Toc37864958)

[5.1 确定存储引擎 5](#_Toc37864959)

[5.2 数据表设计 5](#_Toc37864960)

[5.3 索引设计 5](#_Toc37864961)

[6 数据库实施 6](#_Toc37864962)

[6.1 创建数据库 6](#_Toc37864963)

[6.2 创建表 6](#_Toc37864964)

[6.3 创建索引 6](#_Toc37864965)

[6.4 数据装载 6](#_Toc37864966)

[7 数据库运行与维护 7](#_Toc37864967)

[总结与体会](#_Toc37864968)

1 项目背景

1.设计的原因: 外卖餐饮市场规模扩大逐年扩大，受益于年轻人的线上消费倾向，中国外卖行业不断快速增长。2020年中国外卖餐饮市场规模达到6,646亿元，同比增长15.0%。中商产业研究院预测，2022年中国外卖餐饮行业市场规模将达9417.4亿元。

2020年受新冠肺炎疫情影响，我国整个餐饮行业的收入出现近八年以来的首次下滑，同比下降16.6%。然而，与此同时，我国线上外卖却保持上升趋势，餐饮商户的线上外卖业务收入比重也呈现逐年上升态势。

因此我们认为线上订餐，并且外送餐饮配送以及对此进行相应的处理管理会有巨大的市场。

2.目的：经过多年发展，餐饮管理已经逐渐由定性管理，进入到重视定量管理的科学阶段。众所周知，在定量管理的具体实现方法和手段方面，最有效的工具就是电脑管理。随着当下众多在线餐饮平台的兴起，对于许多中小型商户而言最重要的一点如何抓住用户。才能在竞争激烈的外卖餐饮行业中脱颖而出，活到最后。

因此，我们的目的主要是帮助商户进行数据的量化处理，使得商户能够及时的把握信息并制定或者修改相应的正常。

3.解决的问题：本软件在可以帮助中小企业管理后台数据，分析出当前餐饮环境存在的问题从而改变店铺的营销策略，更快的去迎合市场提高顾客的信任和依赖以及店铺的口碑。如：对顾客订单详情的统计，统计出当月热门菜品对此进行折扣或者菜式改良等。本软件还以针对顾客的评价，对与服务人员以及厨师的评分来决定员工工资以及人员的变更，从而提高店铺对人材管理的利用和收益。

2 需求分析

**2.1 信息要求**

包含的信息如下：1.用户管理权限。2.服务评价管理。3.员工信息管理。

4.订单管理。5.商品在线管理。6.店铺销售统计。

联系：1.用户管理能够决定查看的信息等级。员工信息管理需要服务评价进行信息的补充，商品决定售卖的品类，订单需要从商品中获取售卖信息。店铺销售从订单中进行统筹。

**2.2 处理要求**

每种信息可能需要做哪些处理（增删改查）

1. 用户管理权限能够增加密码账号，删除密码账号，改变密码账号，查询密码账号
2. 员工信息：增删改查：员工编号，姓名，性别，年龄，入职时间，员工地址，员工职位，员工工资，权限。
3. 评价：增加删除修改评分
4. 增删改查订单
5. 增删改查商品
6. 增删改查店铺销售

**2.3 安全性和完整性要求**

有什么安全要求，是否需要分不同用户不同权限

有哪些实体完整性、参照完整性、用户定义完整性要求（大致说一下）

有信息查询和修改信息的安全需求，会按照是否是管理员账号决定是否给予查看和修改一些信息的权限。

实体：用户，员工，菜品，订单，服务。

3 概念结构设计

**3.1 确定实体**

分析提取实体，明确实体之间的联系，实体包含的属性

实体：

1．管理员(属性：管理员编号，管理员账号,管理员密码，管理员等级，管理员姓名)

2．员工(属性：员工编导，员工姓名，员工性别，员工年龄，员工入职时间，员工地址，员工职位，员工评分，员工工资)

3. 菜品（属性：菜品编号,菜品名，菜品价格，菜品口味,菜品售卖)

4. 顾客(属性：顾客编号,顾客姓名,顾客地址)

5. 账单(属性:订单编号.订单时间，订单总价)

6. 订购(属性:订单编号,编号,订购数量)

7. 评价(属性:员工编号,订单编号,评价分)

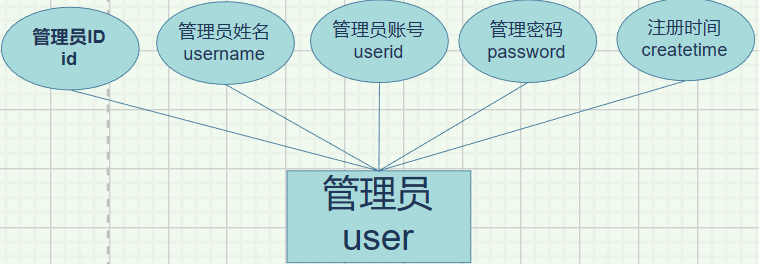
8. 业绩（属性:业绩时间,业绩收入）

关系：

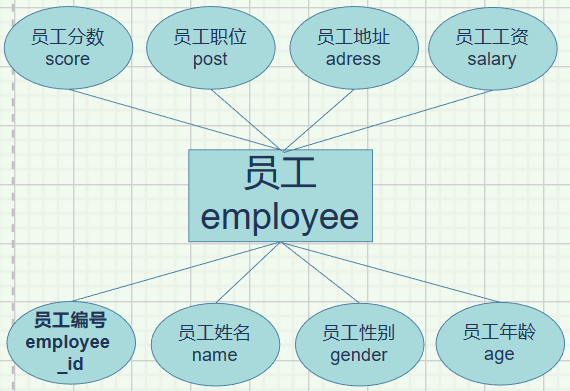
1. 管理员 管理 员工
2. 员工 修改 菜品
3. 员工 制作 订单
4. 顾客 下单 账单
5. 账单 订购 菜品

**3.2 局部E-R图设计**

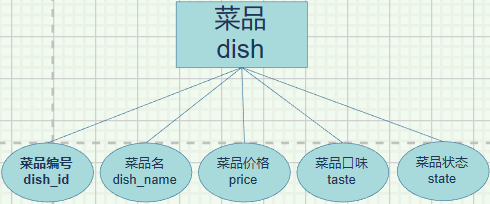
**1. 管理员:**



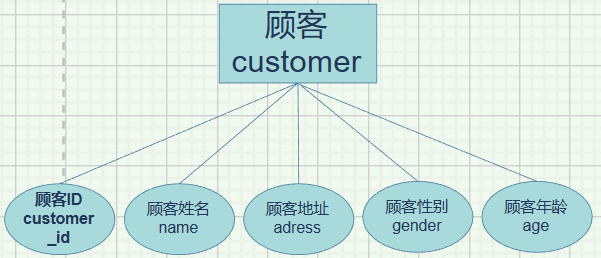
**2.员工:**



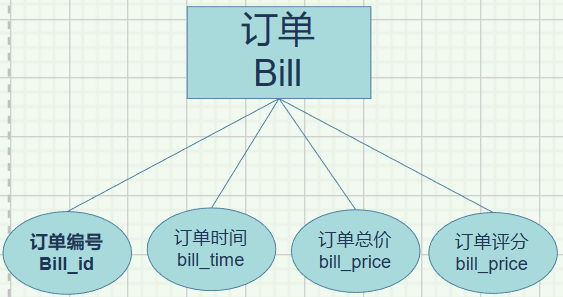
**3. 菜品：**



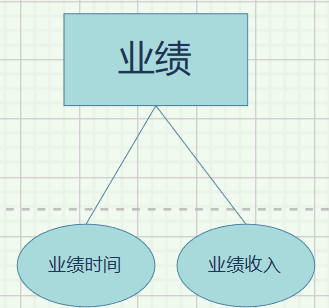
**4.顾客:**



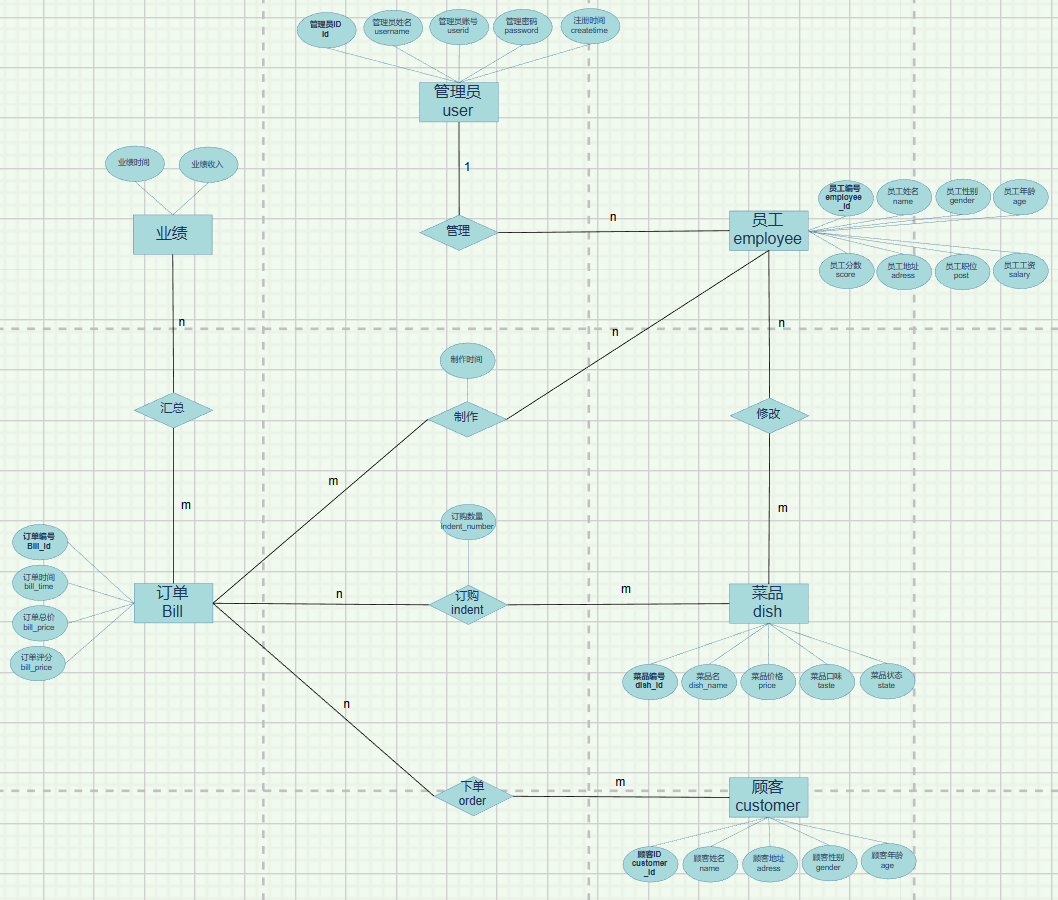
5.订单



**6.业绩**



**3.3 全局E-R图设计**



4 逻辑结构设计

**4.1 关系模式设计**

管理员(管理员编号,管理员姓名,管理员账号，管理员密码，管理员账号创建时间，管理员权限等级)

员工(员工编号,员工姓名，员工年龄，员工性别，员工地址，员工职位，员工工资，员工分数)

菜品(菜品编号，菜品名，菜品价格，菜品口味，菜品在售状态)

顾客(顾客编号，顾客姓名，顾客地址，顾客年龄，顾客性别)

账单(账单编号，账单时间，账单总价，账单评分)

下单(顾客编号，账单编号)(外键:顾客编号，账单编号)

订购(订单编号，账单编号，菜品编号，订购数量)(外键:账单编号，菜品编号)

制作(员工编号，订单编号，制作时间)(外键:员工编号，订单编号)

业绩(业绩编号，业绩时间，业绩收入)

**4.2 关系模式优化**

5 物理结构设计

**5.1 确定存储引擎**

使用Innodb,理由：它遵循ACID模式设计，具有与事务（Transactions）、回滚和保护用户数据的崩溃恢复能力。

InnoDB引擎在主内存中维护了一个buffer pool，用来缓存表和索引，频繁访问的数据会直接在内存中操作。

设计表时为每个表添加了主键，会自动优化涉及列的操作。也就是说基于主键的查询、排序、聚合性能会很高。

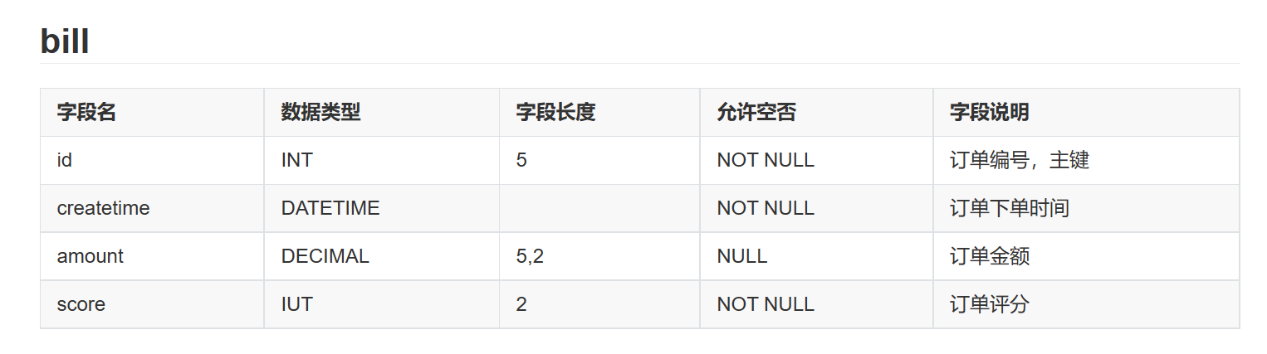
InnoDB提供行级锁，拥有与Oracle类似的不加锁读取。InnoDB锁定在行级，并在SELECT语句提供一个Oracle风格一致的非锁定读。这些特色增加了多用户并发性和性能表现。

InnoDB表基于主键在磁盘上安排数据，有优化的

为了维护数据完整性，InnoDB还支持外键完整性约束。

InnoDB可以与其它MySQL存储引擎混合使用InnoDB表，即便是在同一个SQL语句中也可以使用多种存储引擎的表。

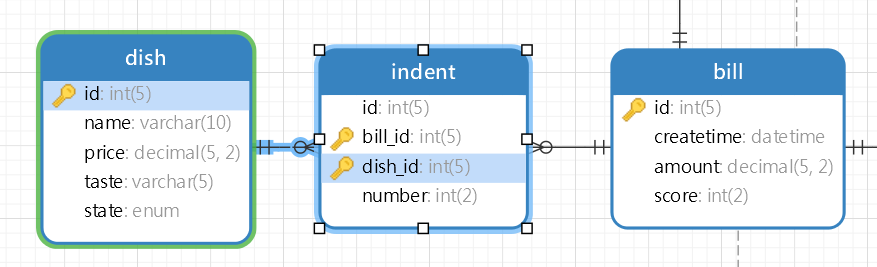
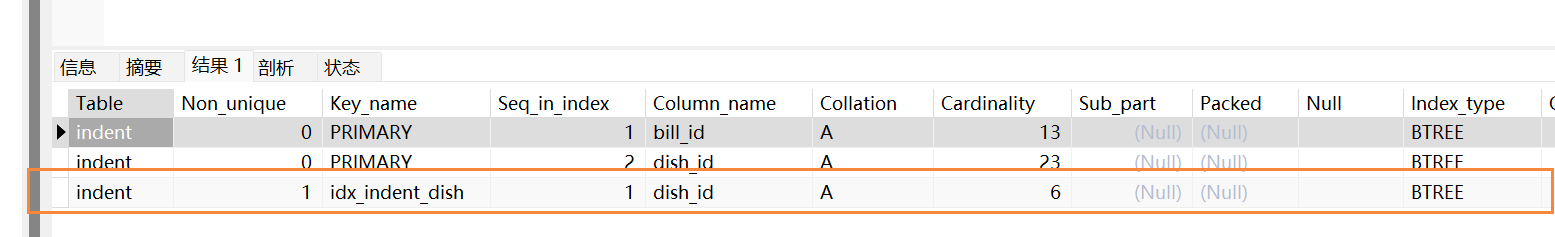
InnoDB的设计初衷是为了在处理大量数据时提高CPU效率和性能。InnoDB可以存储大量的数据，即使操作系统限制单个文件大小为2G的系统上也如此。

* 1. **数据表设计**



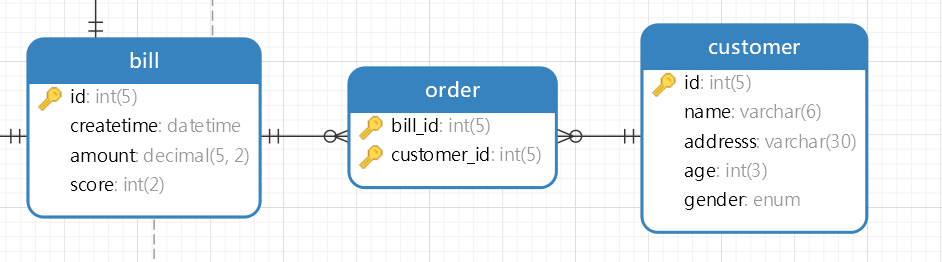
**5.3** **索引设计**

* 索引



建立有关菜单（dish）和订购信息（indent）的索引

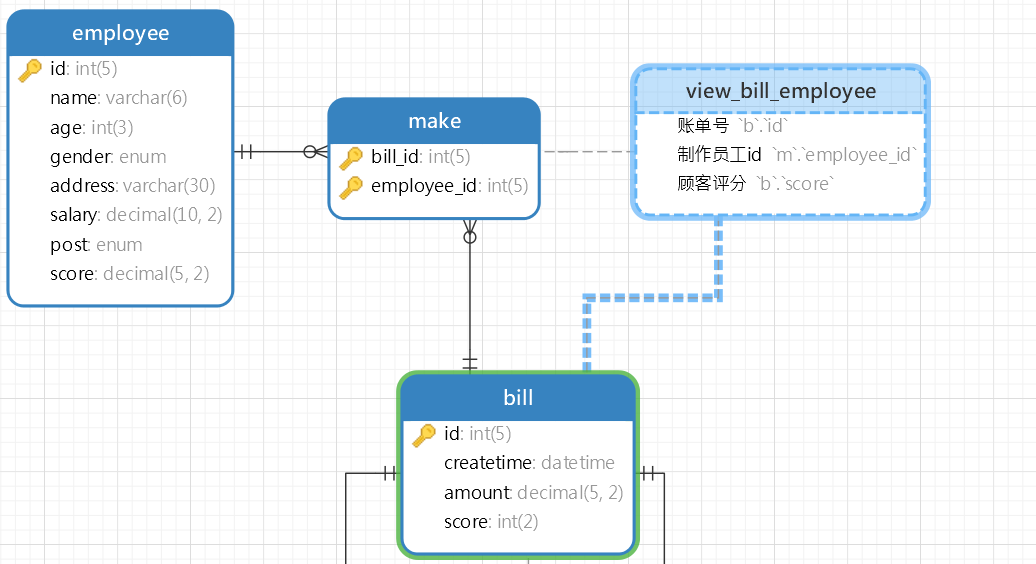
便于账单号（bill.id）查询到指定订单(indent. id)里的具体产值账单对应菜名(dish.name)



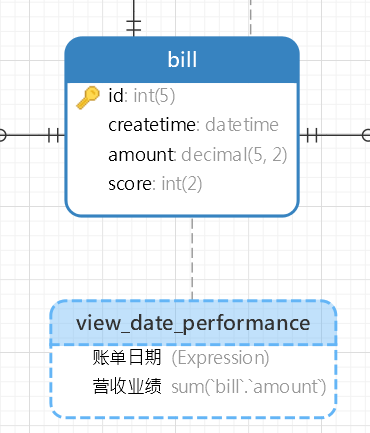
建立有关顾客（customer）和下单（order）的索引关系，

便于通过账单查询到指定订单号下的顾客姓名（customer.name）

* 视图



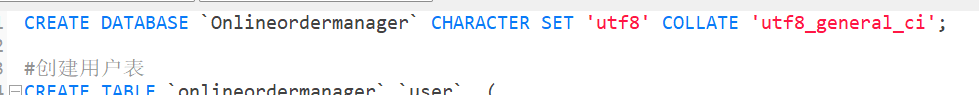
该视图统计出 员工employee表id和对应其制作(make)的账单号（bill.id）和账单评分(bill.score)

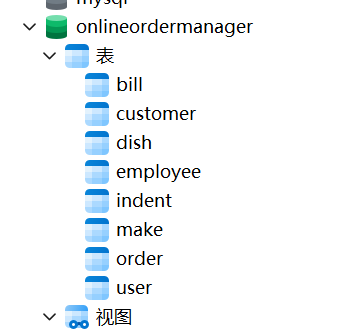


该视图有统计了每天总计的营业额，便于管理员查看到具体每天的营收业绩

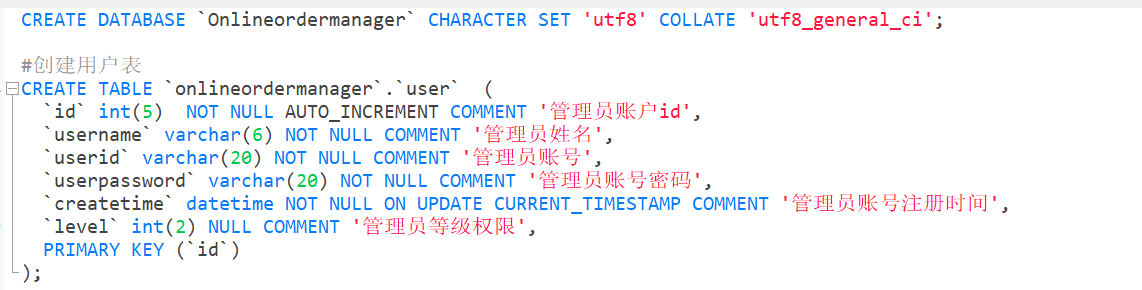
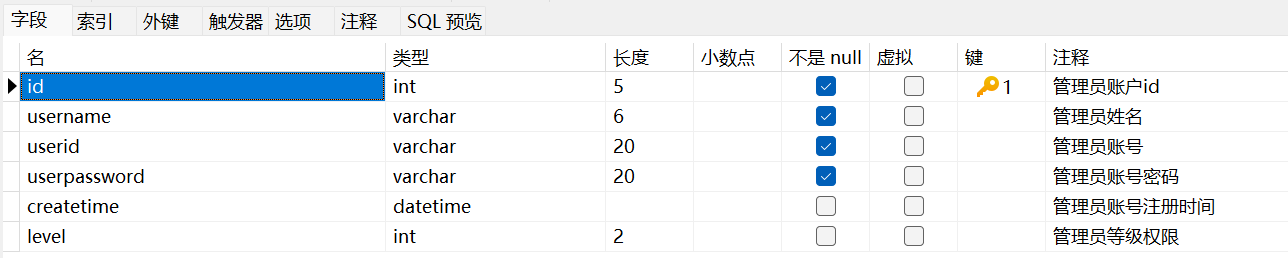
6 数据库实施

* 1. **创建数据库**

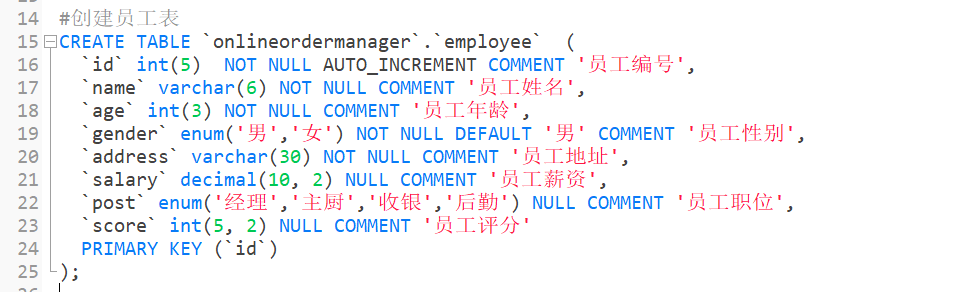


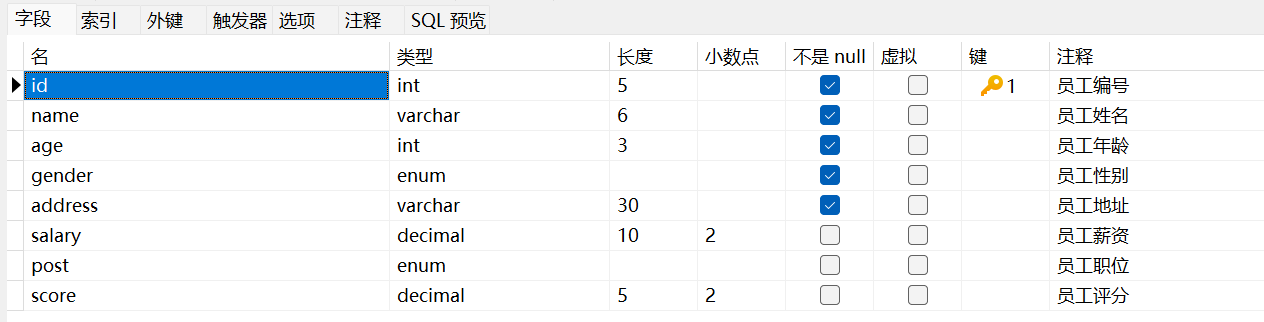


**6.2 创建表**

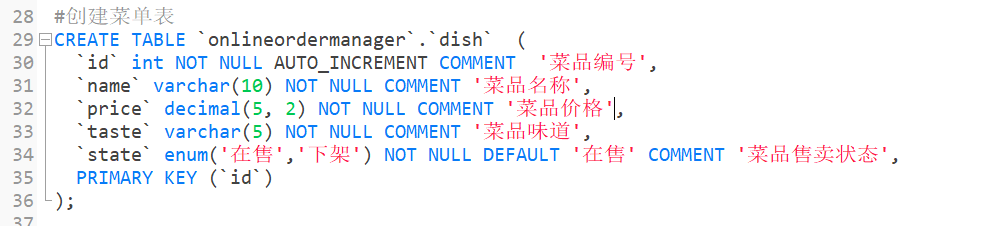
**创建用户表**

**创建员工表**



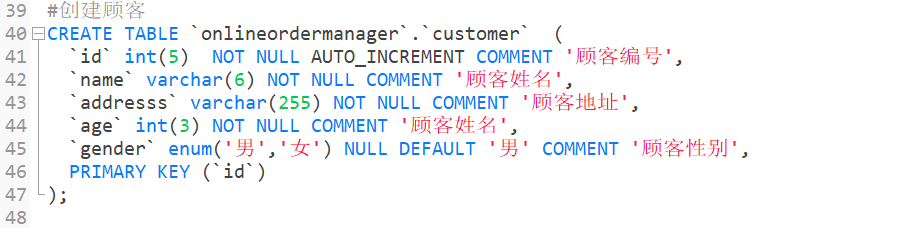


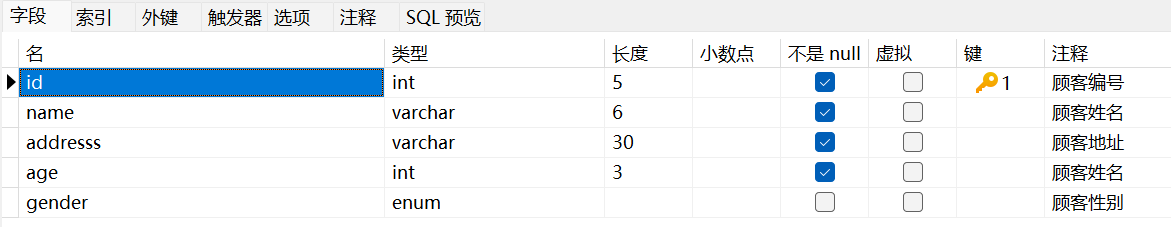
**创建菜单表**



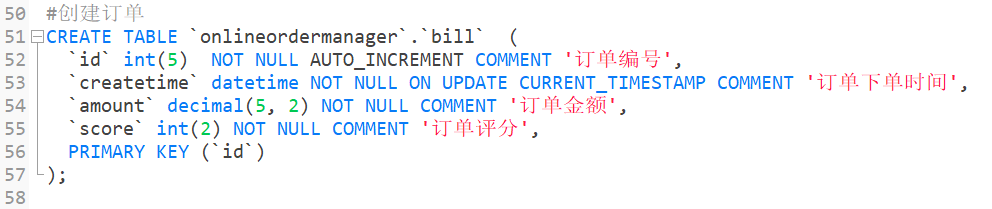


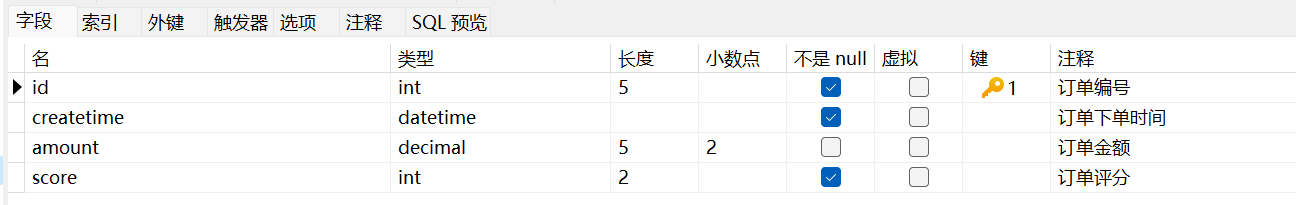
**创建顾客表**



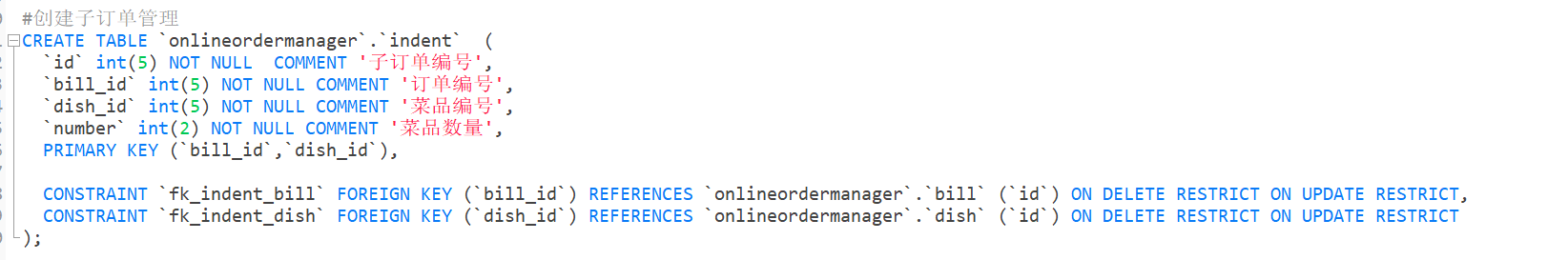


**创建账单表**



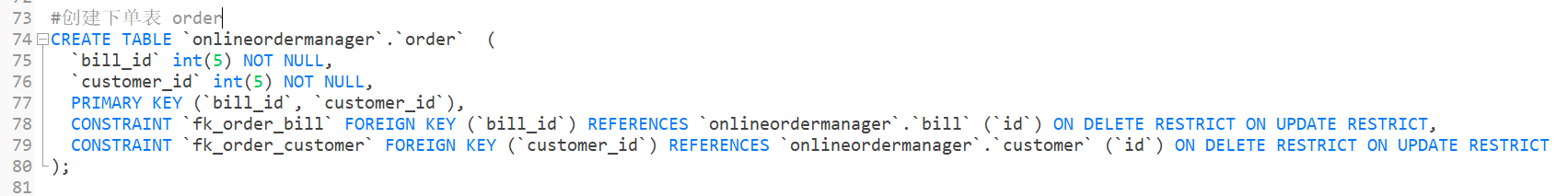


**创建订单表（关联账单和菜品）**



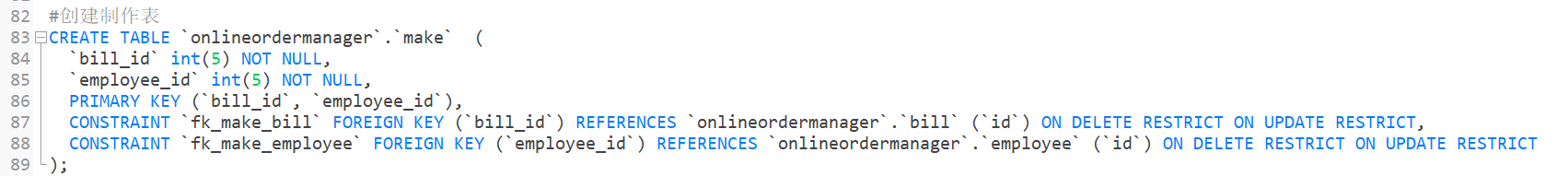


**创建下单表（关联顾客和账单）**





**创建制作表（关联员工和账单）**





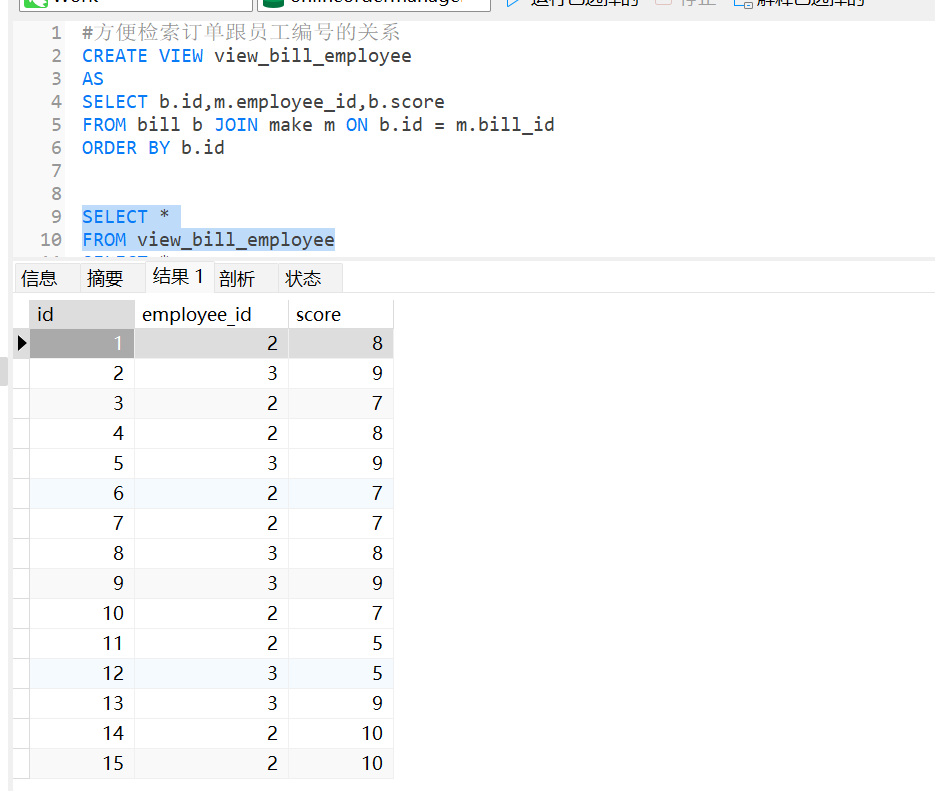
**6.3 创建用户**

**6.4 创建索引**

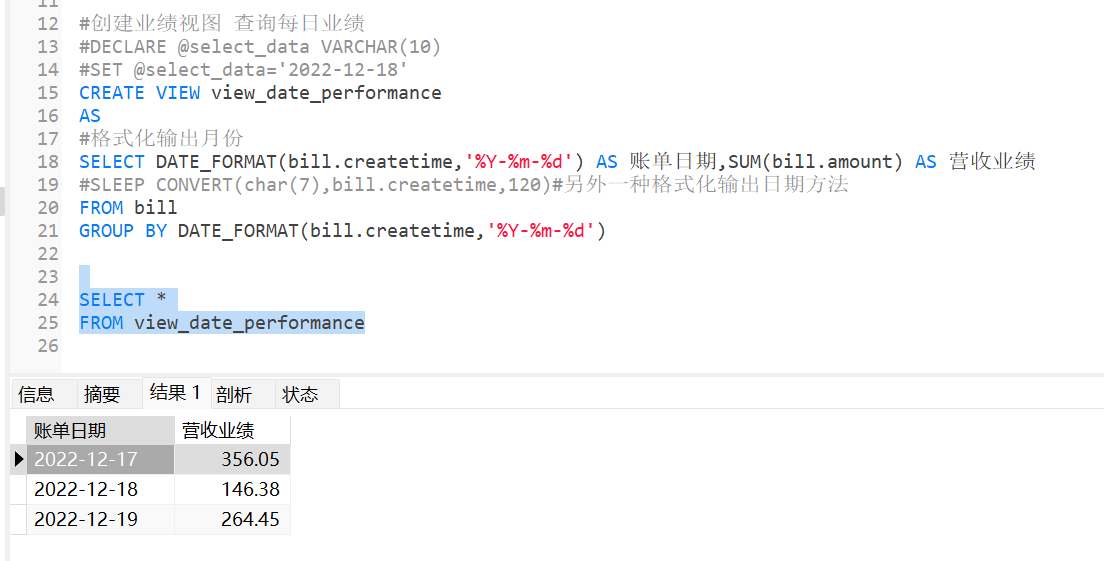
1、创建员工订单视图，方便管理查询订单制作人员。

有利于代码利用，有利于触发器的查询

Ps: 7、数据库运行 里第8点有用到



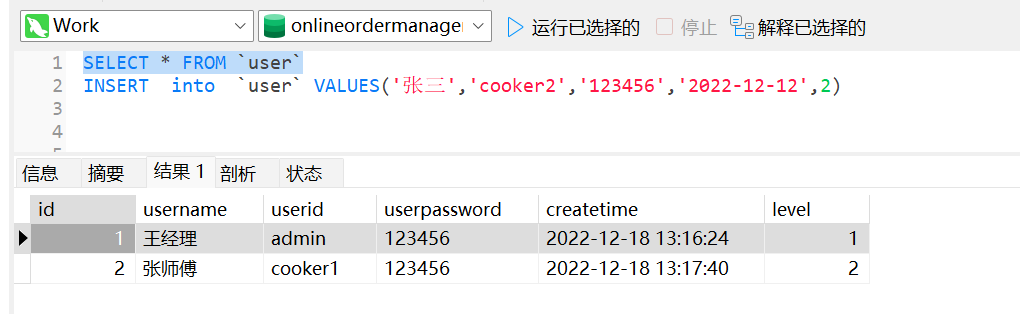
**2、每日业绩视图汇总**

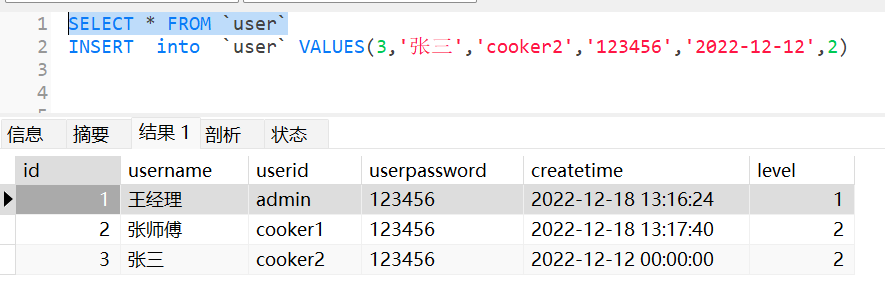


根据天数进行分组汇总出每天的营业金额

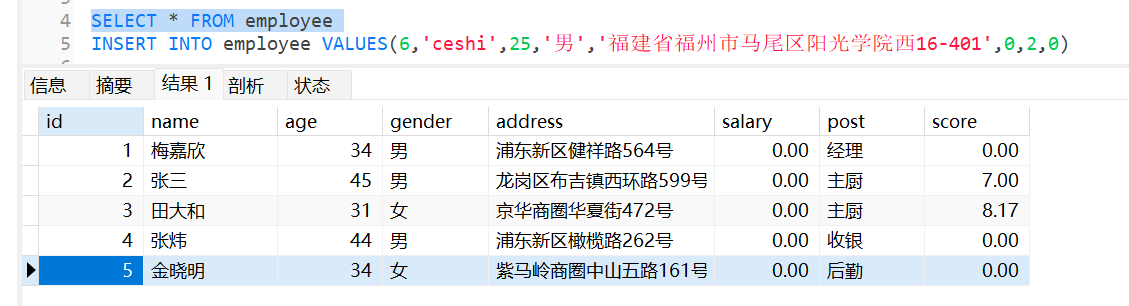
**6.5 数据装载**

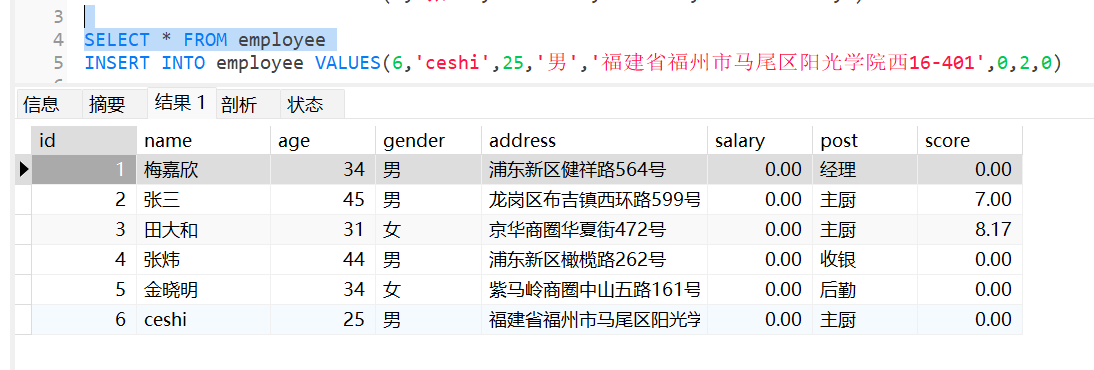
1、用户表数据插入(图1为插入前，图2为插入后)





2、员工表数据插入



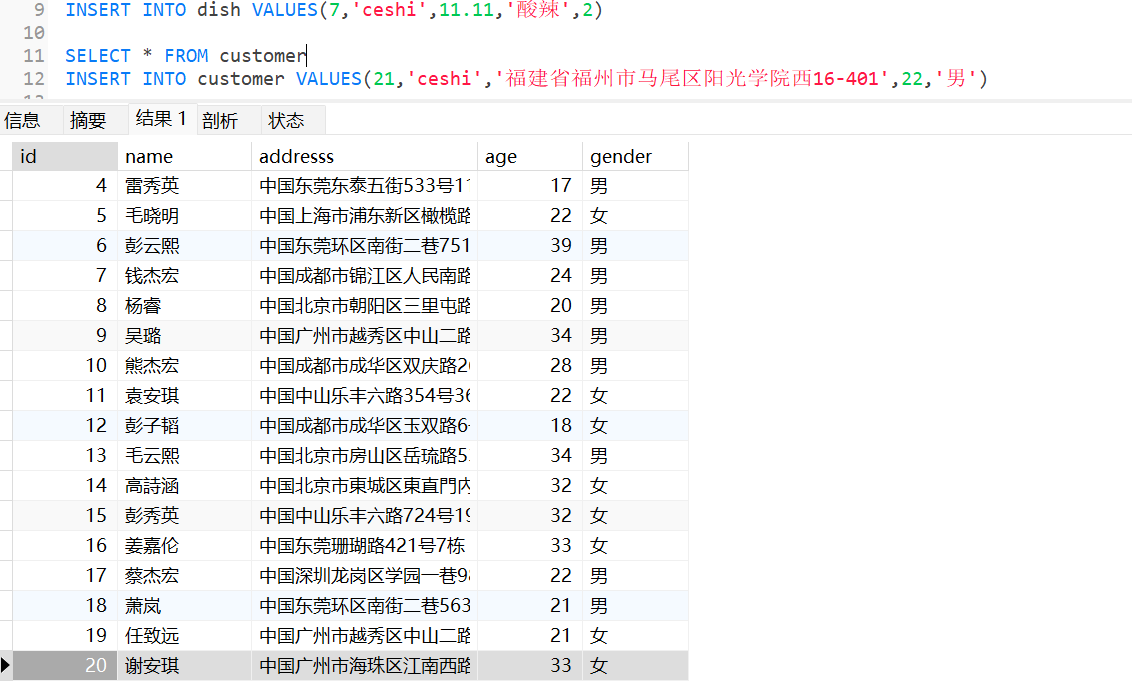


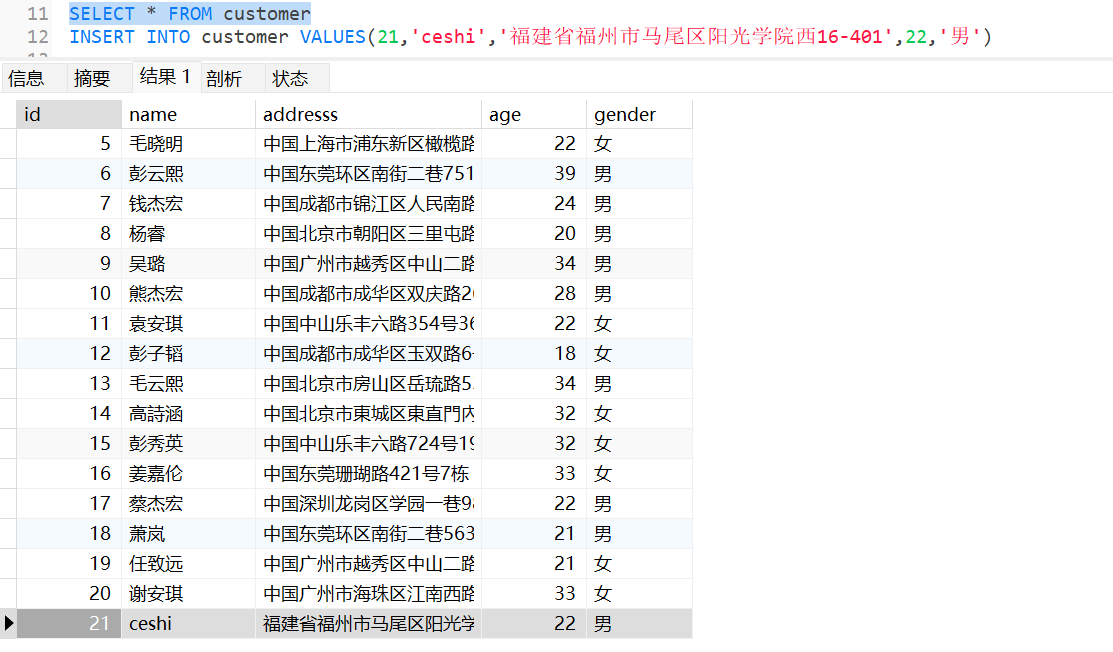
3、菜单表插入



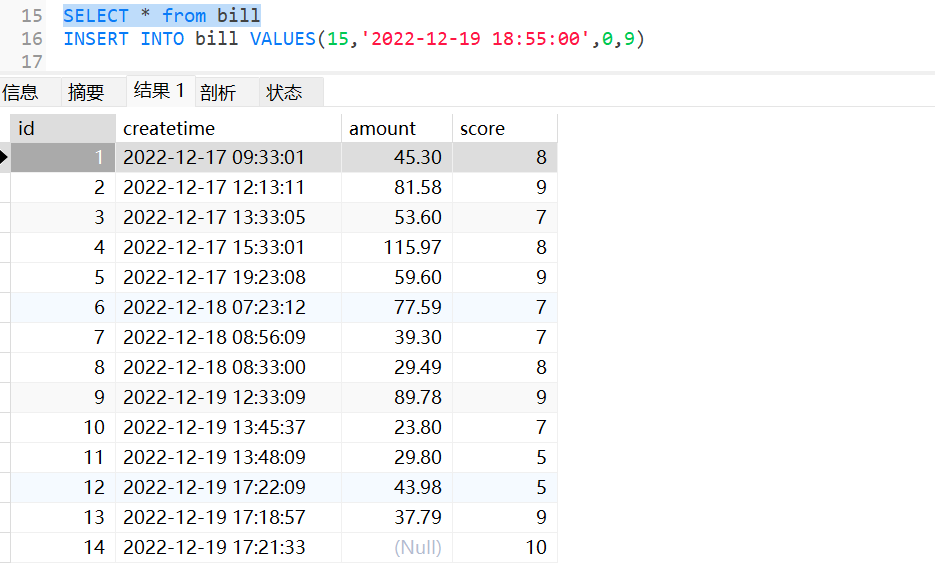


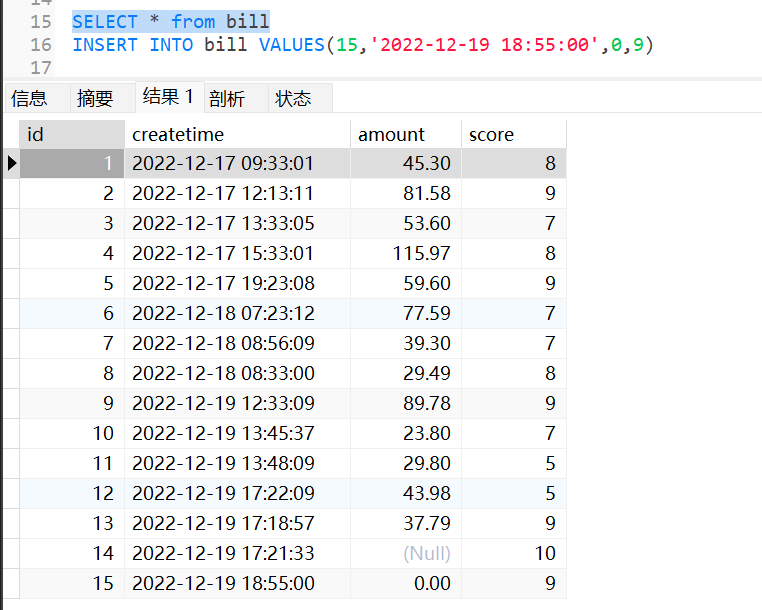
4、顾客表插入





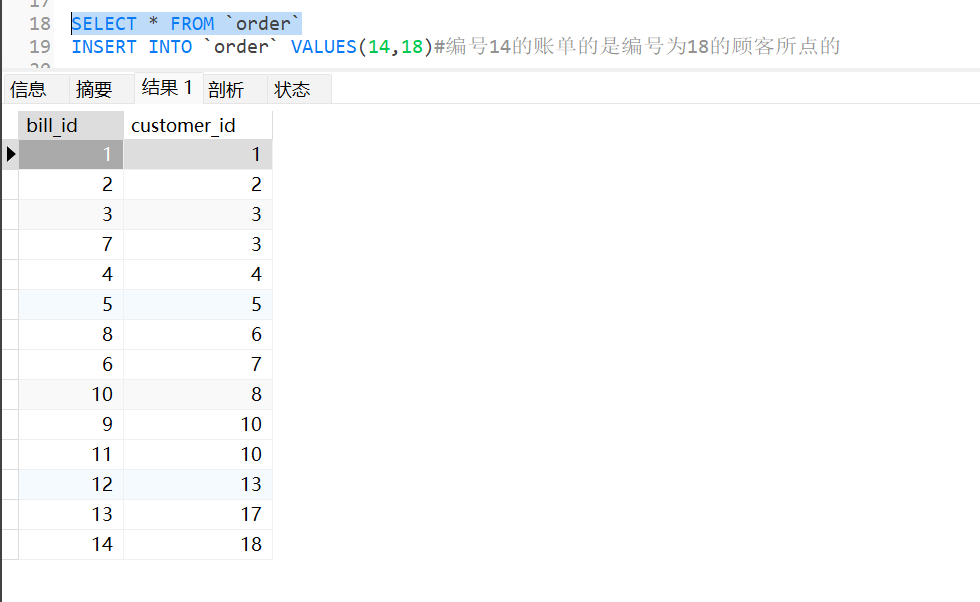
5、账单表数据插入



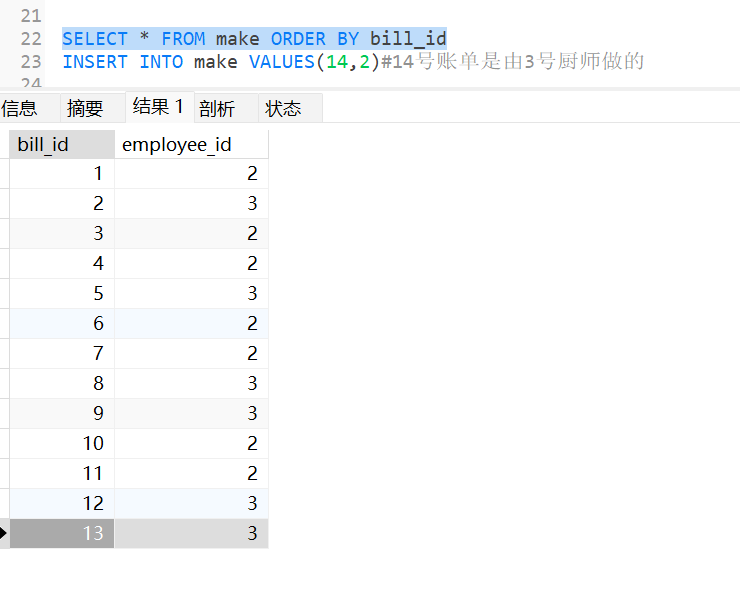


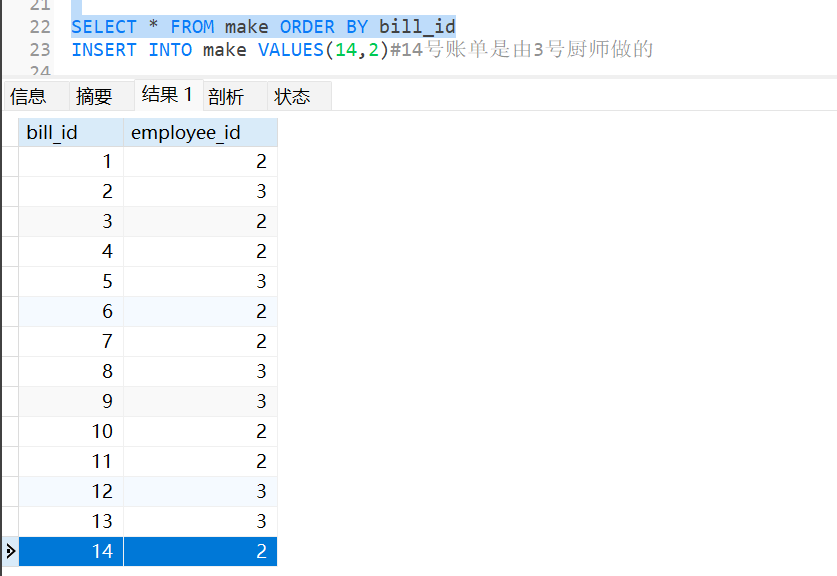
6、下单表数据插入

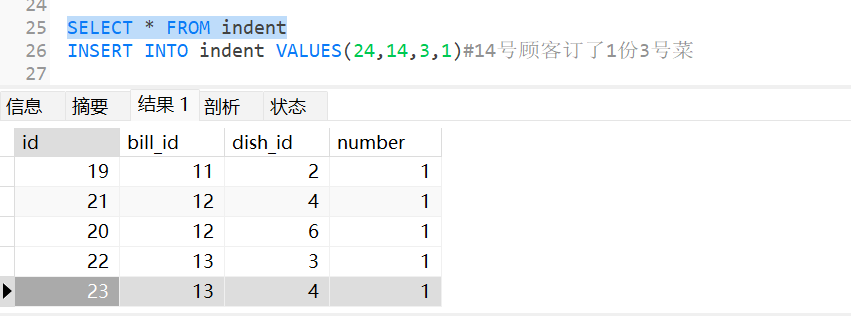




7、制作表数据插入



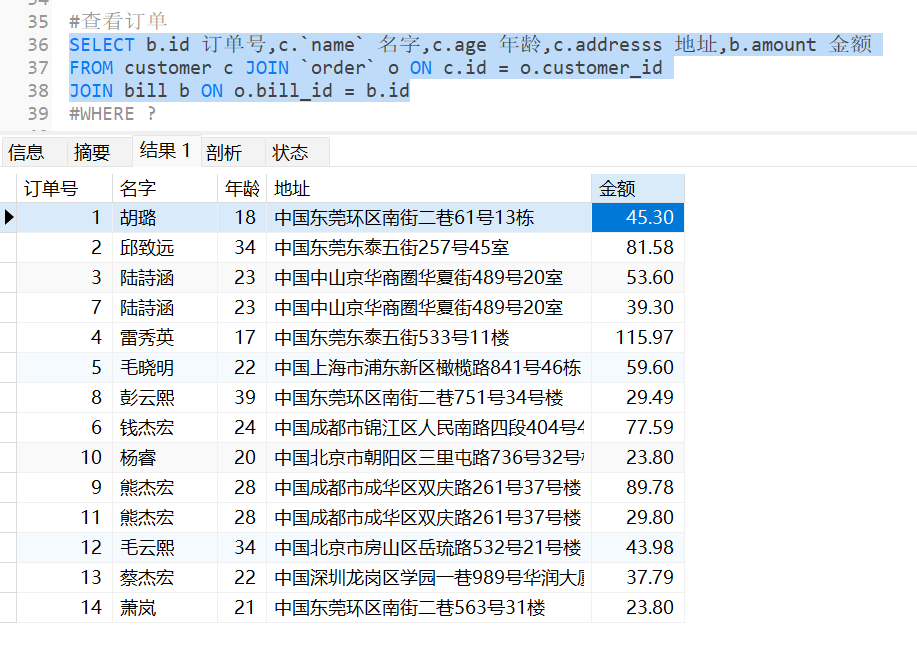


8、订购表数据插入



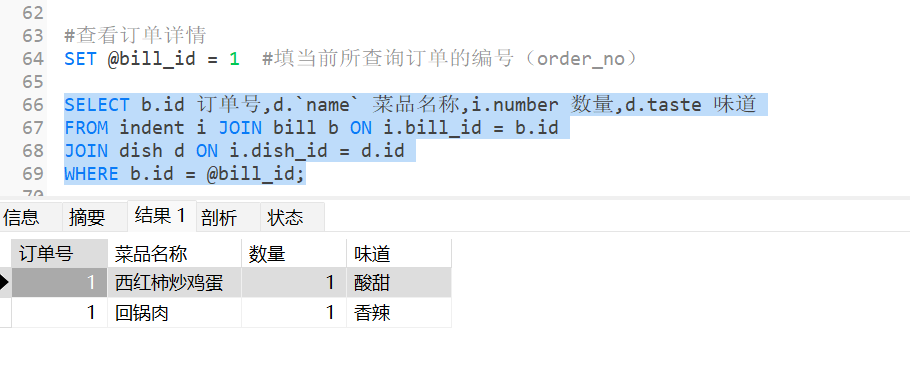
7 数据库运行与维护

1、查询所有顾客的订单



可以查询用户订餐的大致情况

2、查看具体订单号的详情



可以查看给出指定单号的具体订单内容：

如图查出订单号为1的订单详情

3、对用户订单进行分页查询



可以对用户的订单进行分页查询，设置分页变量和存储过程，方便小组成员在搭建网页时 进行数据调用

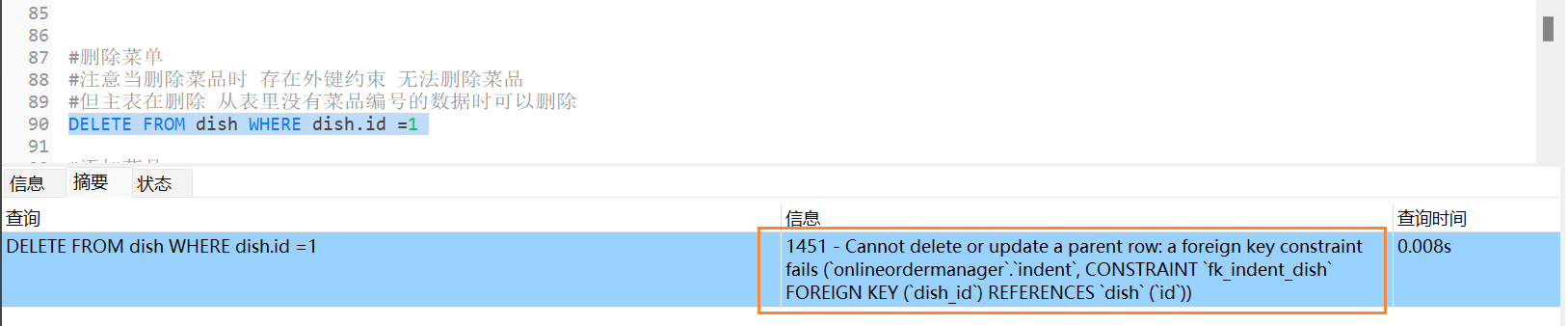
4、修改菜单



可以将菜品名字修改，价格修改，味道修改同理

5、删除菜品

尝试删除菜品编号为1的菜



删除失败，数据时存在外键约束的数据存在从表里 则无法删除



尝试删除菜品编号为7的菜品

删除成功，对于删除从表里没有的数据时 可以删除



6、修改员工的工资



将所有员工的工资设置5000



修改成功

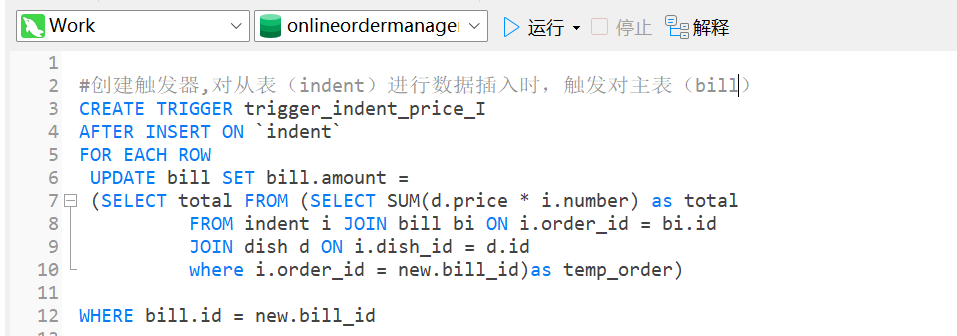
7、查询每日综合业绩



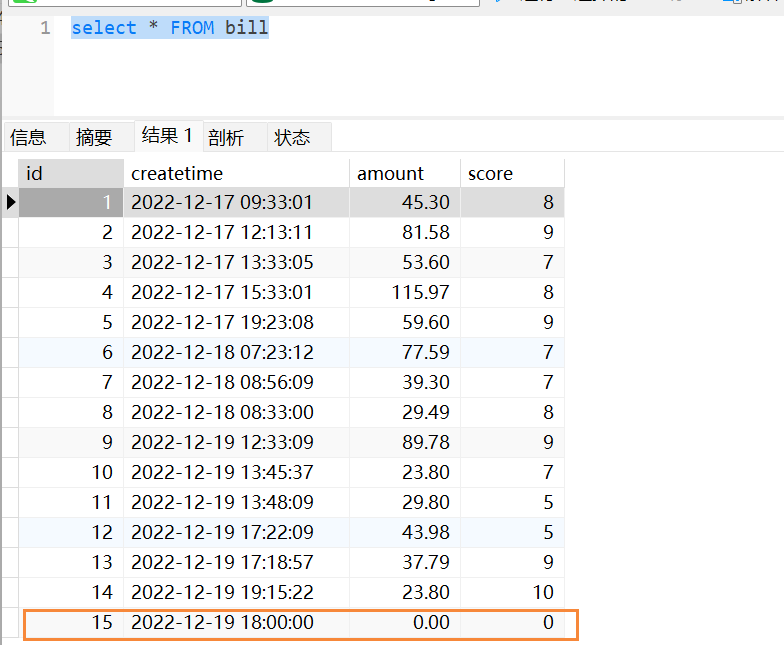
需要将下单时间datetime数据格式进行转化为datetime,然后按照时间进行分组，统计出每日的营收业绩

8、订单金额数据汇总

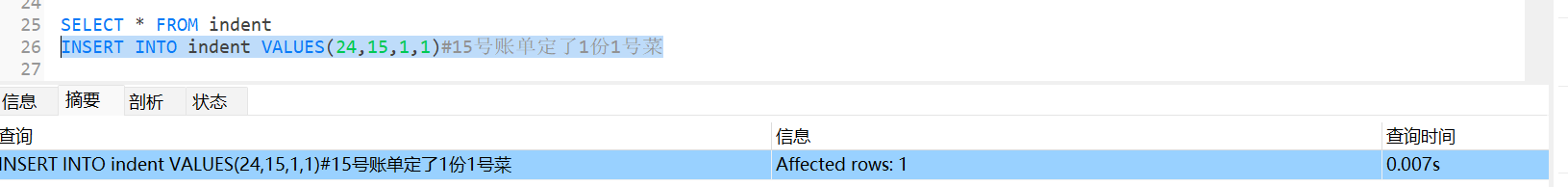
创建触发器trigger\_indent\_price\_I



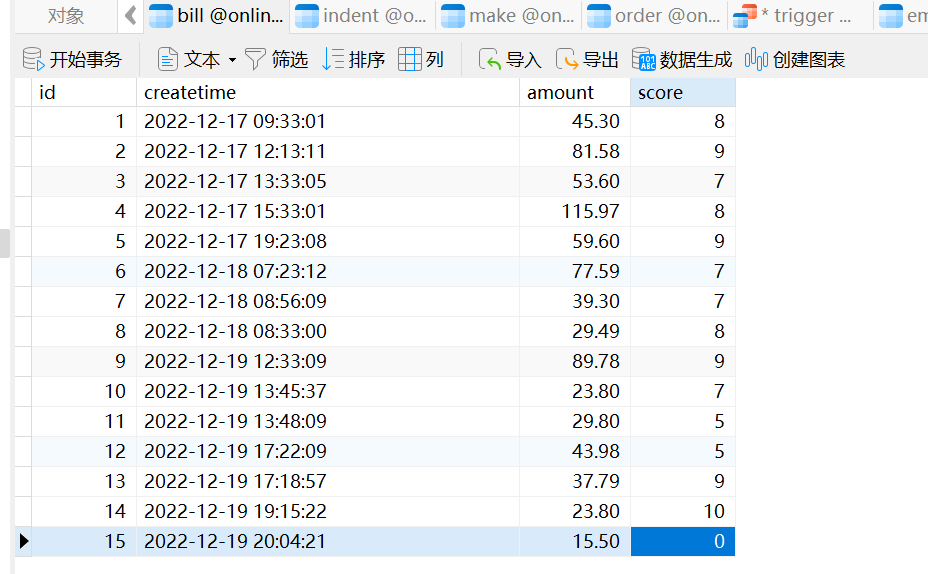
当对intent指定用户的具体订餐情况时就会触发该触发器，将该批订单号的所有菜品价格汇总更新到bill的amount字段上。



查询到编号为15的账单并没有统计价格，我们对intent录入顾客的具体点餐情况



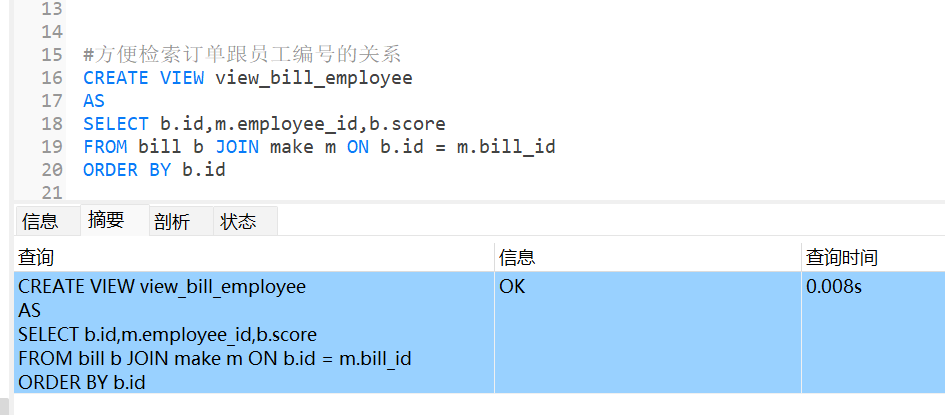
直接查看账单表，可以看到数据已经汇总到主账单里面了

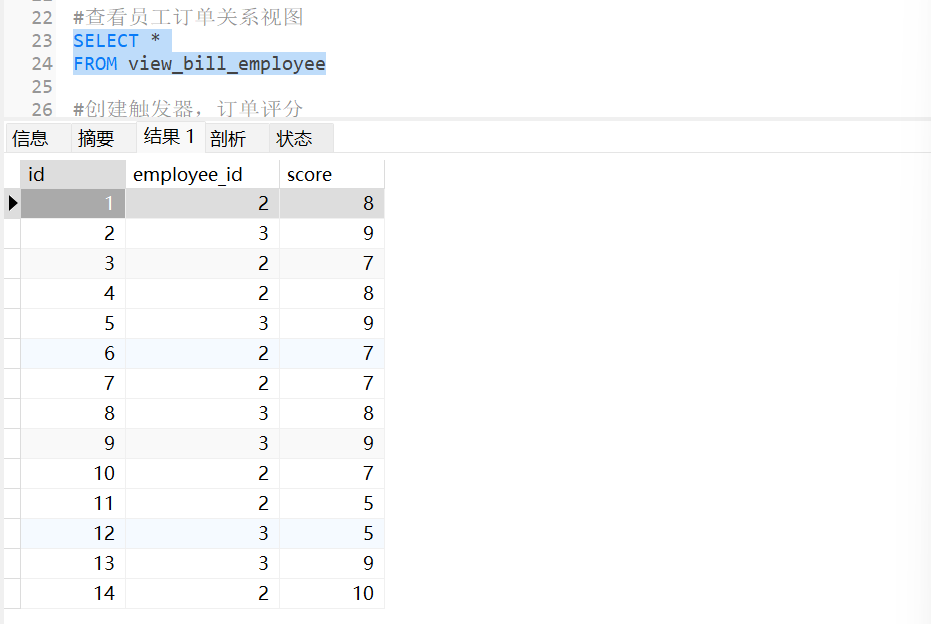


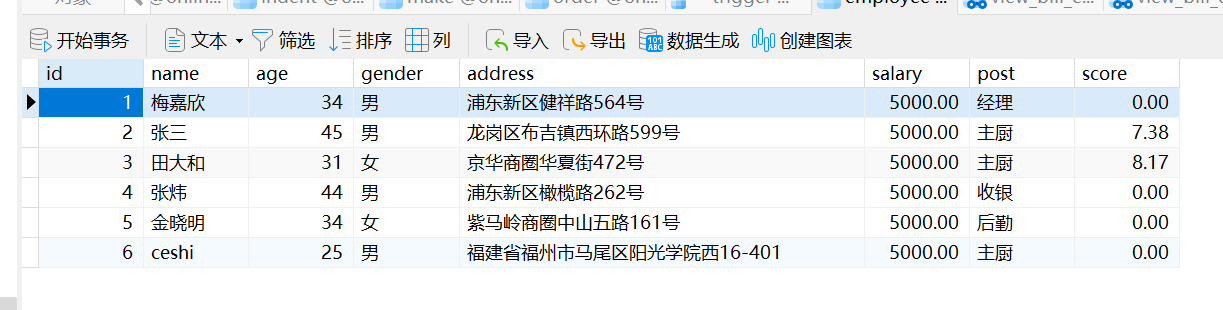
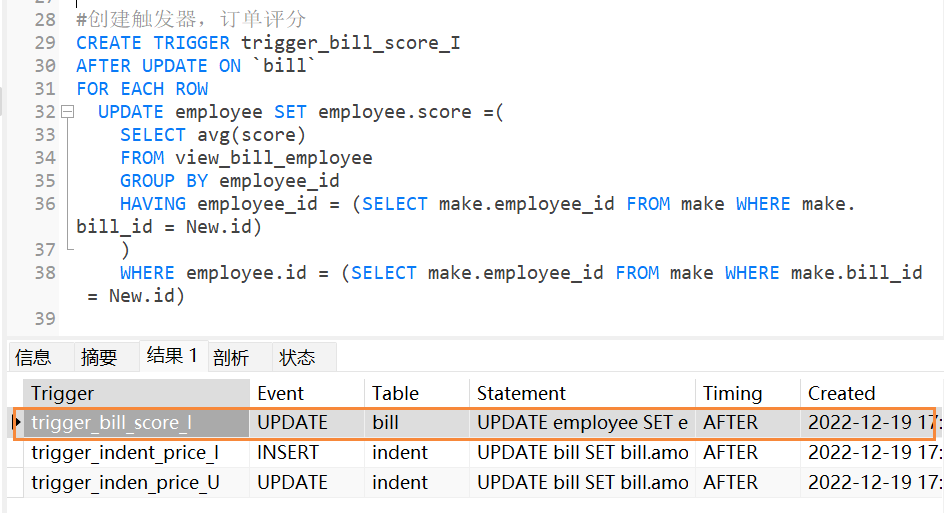
价格 统计触发器完成

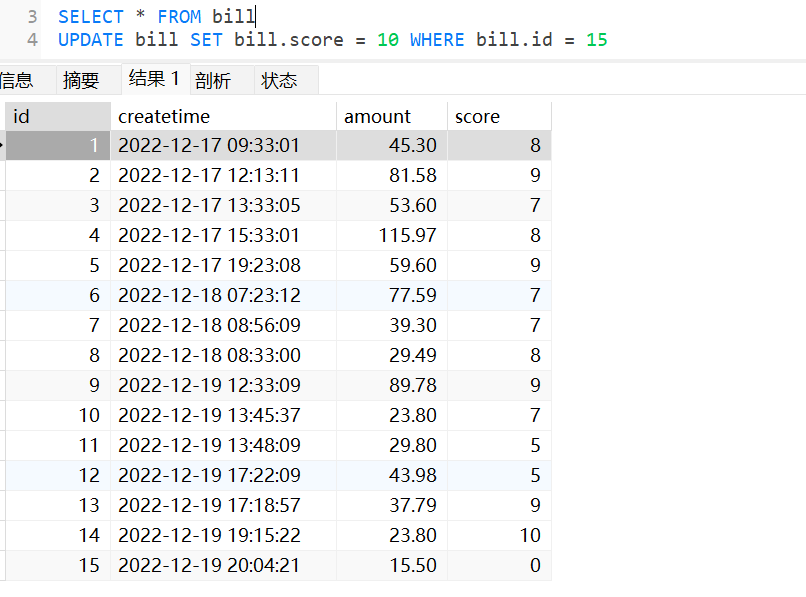
9、员工评分触发器

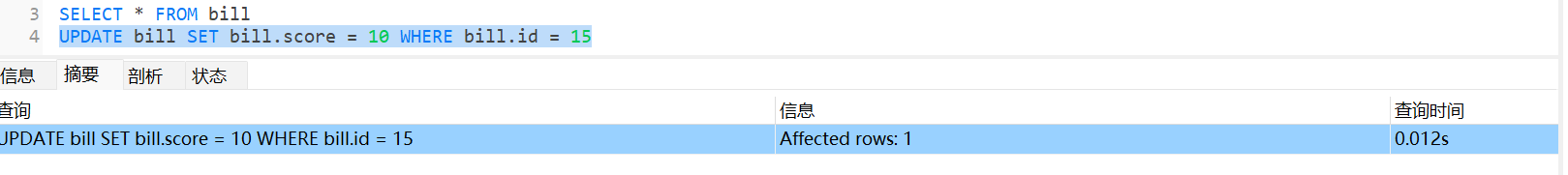
每当一个订单生成，并且用户完成评价时，我们将根据员工和订单的关系生成视图

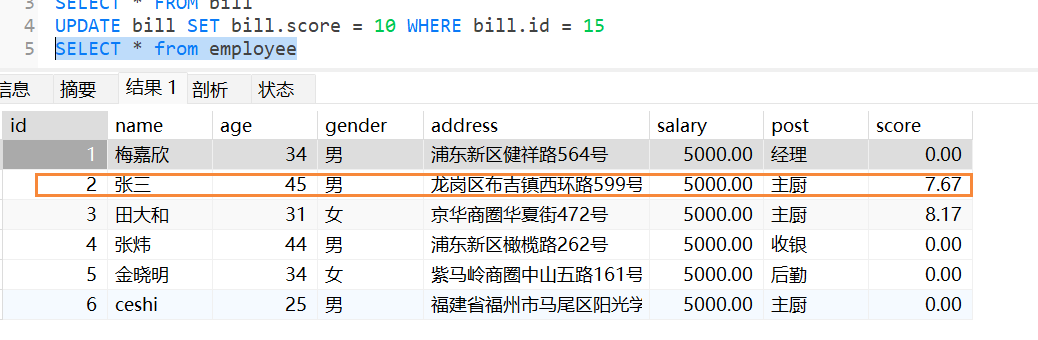




创建触发器

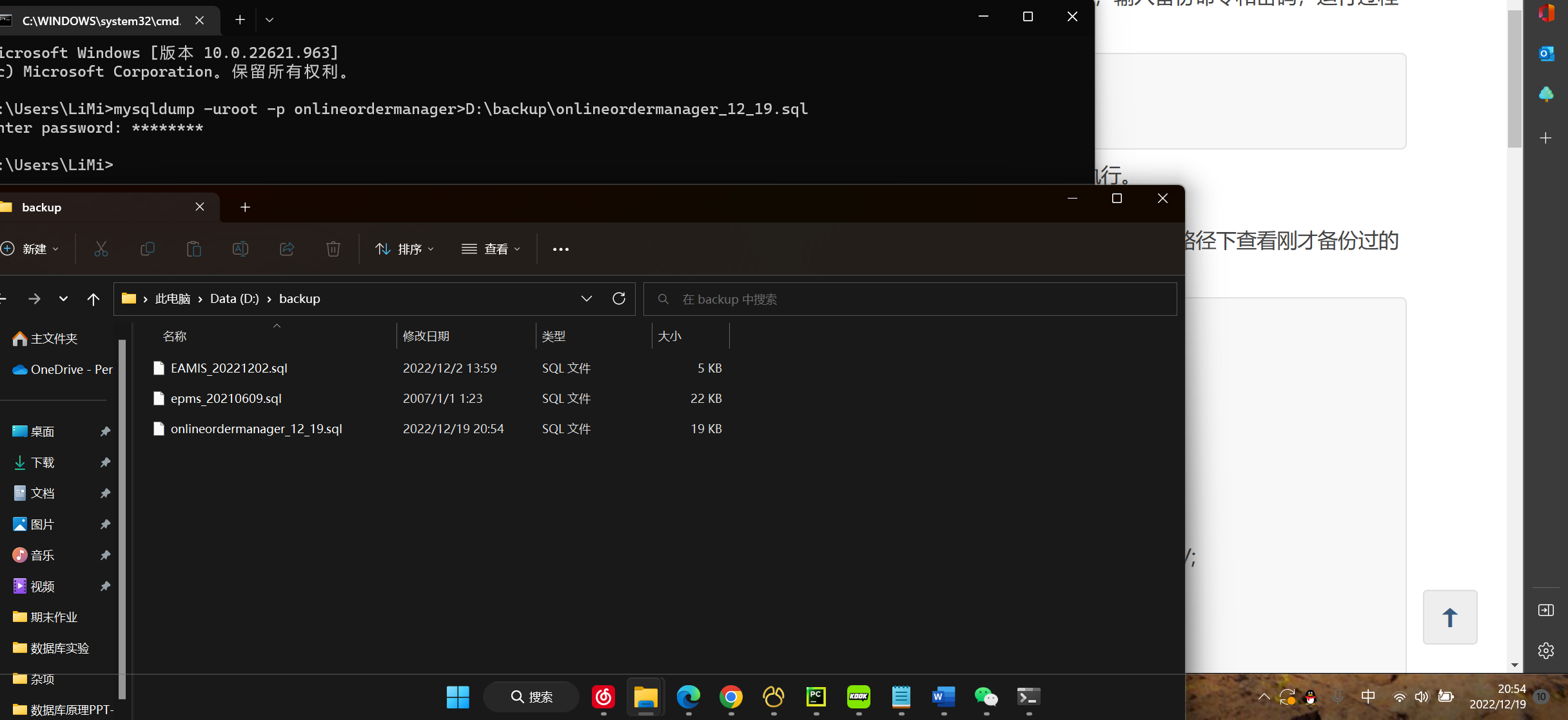
当用户完成评价对bill的score字段进行更新时，将会触发 将通过视图订单取得所对应制作员工的id 进行对该员工的id进行分组算出平均分 ，汇总到员工employee.score上





触发完成

10、数据库备份



备份完毕

总结与体会

李弦：

这次的数据库实验项目中我完成了触发器和一些查询语句包括一些关系转化实体的表的创建如order,intent。

在模拟顾客订购点单时，所插入和更新的数据，会汇总各个相关的表中，

保证了数据的完整性

我设计的数据库的触发器中遇到了一个头疼的问题，触发触发器时，不知如何拿到新插入的数据，为此我查阅资料，

找到了两种解决办法

* 查找临时表

在sql数据库中有隐藏的 名为inserted和delected的临时表，它们存放在内存中，是临时表。当触发器工作完成，它们也被删除。它们是只读表，不能向它们写入内容。

1.插入操作（Insert）

inserted 表有数据，deleted 表无数据

2.删除操作（Delete）

inserted 表无数据，deleted 表有数据

3.更新操作（Update）

inserted 表有数据（新数据），deleted 表有数据（旧数据）

* New.字段名

在了解到 上面那些后，我又从mysql官网手册的5.1的版本中了解到 新插入的数据可以用 New.字段名 来获取到刚刚插入或者是更新的那一行元组的字段值。

在这里我才用的是第二种方法解决的。

许锦炜:

在这次项目中我完成了表的关系模型转换和需求分析,同时完成了一些后端操作。本人在项目中涉及到了web界面的后端程序设计，主要复制数据库的数据查询和数据更新内容，将web页面中所需要的数据通过后端程序算法将mysql中的对应数据查询并封装，传递给前端设计。

数据库层面主要为基础程序设计提供建议，与后续部分SQL语句的编写和改进。这个数据库主要用于外卖行业的数据处理功能，将日常在店面中销售的外卖数据采用web网页的形式显示出来。方便企业用户的数据操作。

在本次团队项目中，我学到了很多，例如

1.团队项目中如何在前期设计数据库，以及后续项目如何有序的跟进。

2.前期数据库设计中，数据库的耦合度太高，导致数据冗余问题，在我们进行分解后，数据表的冗余度有了明显下降。

3.在进行后端设计中，SQL语句返回出来的数据类型，与GO语言结构体的数据类型不一致问题，导致前期的数据频繁报错，后面采用自建结构体数组进行封装后解决。

吴嘉骑：

我在这次的实验项目中，创建了数据库，丰富了字段名，规定了字段的数据类型。完成数据库的基本建设，在后期也同小组成员，一同分析，解决了某些的表明模糊和字段含义模糊的问题，为了方便成员理解和沟通。

在建表中遇到了一个小问题，对于在从表中设置外键时，主表的主键本身时自增id,从表没有将外键设为自增键，而无法创建该表，从而导致报错。和组员讨论分析后成功解决了这个问题。完成了实体的约束。

肖冰雁：

对于这次的实验我完成的是概念结构设计和表的字段完善，确定了基本的实体，完成了局部er图和全局er图。在明确我们小组的目标后我就着手去收集了相关材料，了解到在现暂以管理平台随着今年些年外卖激增，同时商户也是络绎不绝开启了线上订单购物。基于此我和小组成员沟通后，第一对于用户的管理权限做出了明确的划分，第二 对于实体间的相互关系的完善，在最初对顾客订单和菜品的三元关系 实际上并不合理，随后我将三元关系装换成了，两两之间的实体关系，优化了表的结构。第三初步拟定了数据库具有的功能，例如用户订单反馈的评分，每日营收业绩分析。

代鸿杰：

通过这次参与数据库小组项目，我主要完成了语句的查询和表之间的约束关系，同时完成前端的页面的优化和数据调用。

在前端中出现数据调用不合理的地方及时和组员进行沟通进行数据类型的优化和修改。

再对于菜品表的删除处理时，一开始由于没有对从表进行外键约束，从而导致菜单号丢失，用户的账单记录里查询不到顾客具体订餐数据，再对于indent表进行外键约束，当从表里有使用的数据时，主表无法对该数据进行删除，满足我们对数据记录的合理规划。当使用级联删除时虽然可以完成菜品的删除，但是会造成我们订单记录的丢失这里，没有采用级联删除。