Lewis University  
CPSC 50900: Database Systems  
Term Project

Commodity possession in relationship with employment

Pranay Reddy Lingapuram, pranayreddylingapu@lewisu.edu

Srinivasa Rao Medisetti, smedisetti2806@lewisu.edu

https://github.com/Pranay4683

Table of Contents

[Initial Proposal 2](#_Toc85814569)

[Data Sources 2](#_Toc85814570)

[Data Storage Alternatives 2](#_Toc85814571)

[Relational Database Design Process 3](#_Toc85814572)

[Relational Database Design 3](#_Toc85814573)

[Data Definition Language (DDL) Scripts 3](#_Toc85814574)

[Data Manipulation Language Scripts 4](#_Toc85814575)

[Indexes 5](#_Toc85814576)

[Views 5](#_Toc85814577)

[Triggers 5](#_Toc85814578)

[Transactions 5](#_Toc85814579)

[Database Security 6](#_Toc85814580)

[Locking and Concurrent Access 6](#_Toc85814581)

[Backing Up Your Database 6](#_Toc85814582)

[Python Programming 7](#_Toc85814583)

[PHP Programming 7](#_Toc85814584)

[Suggested Future Work 8](#_Toc85814585)

[Activity Log 8](#_Toc85814586)

# Initial Proposal

*Description: You will describe the data you aim to store. What data will be storing? Why are you interested in this data? Why is it important? Where will the data come from? Who will use this data? What kind of application do you plan to build with it?*

*Rubric: Your response to each of these six questions will be graded out of 3 points.*

* *3 points: clear, complete descriptions that convey the importance and meaning of your data*
* *2 points: mostly clear descriptions, although some additional data would have helped in some sections*
* *1 point: necessary details are lacking in many of your responses.*

*You will also earn 2 additional points for coming up with a descriptive title for your project.*

*Total points possible: 20*

We want to store the data of different countries wherein how many people are unemployed and how many people own cars. We want to figure out what is the ratio between having a job and buying a car. The data will come from [Industrial Commodity Statistics Database](https://data.un.org/Data.aspx?q=cars&d=ICS&f=cmID%3a49113-0#ICS) and  [International Labour Organization](http://www.ilo.org/stat/)   
 We plan on building informative application.

Title: Commodity possession in relationship with employment

# Data Sources

*Description:* *Gather your data in text files. The text files may be csv, tab-delimited, xml, json, or some other custom format. Not all the files need be of the same type. Identify what each file contains by indicating where it came from, explaining in detail how it structured, and describing how you will reorganize the data into a relational database. Post your data files to your GitHub repository, and provide samples of the data in your Word doc.*

*Rubric: Your work will be graded as follows:*

* *5 points: you gathered multiple data files that contain the data that will populate your databases. If you do not use multiple data files, you will not receive credit.*
* *5 points: you described the contents of the data files in detail, including referencing their origin and explaining how they were structured.*
* *3 points: you identify which fields you plan to include in your database, including their data types and any constraints you expect to impose on the data or steps you'll have to take to clean up the data.*
* *2 points: you post the data files to your GitHub account and make it possible for me to see them.*

*Total points possible: 15*

Data sources: <https://data.un.org/Data.aspx?q=people&d=LABORSTA&f=tableCode%3a3A%3bcountryCode%3aLA#LABORSTA>

[*https://data.un.org/Data.aspx?q=cars&d=ICS&f=cmID%3a49113-0*](https://data.un.org/Data.aspx?q=cars&d=ICS&f=cmID%3a49113-0)

*Data files contain records of number of cars bought by countries in respective years and how many people were unemployed during that time period*

*Data of car contains number of units sold(int), country(string), year(date), value(int)*

*Data of unemployment file contain country(int), year(date), sex(string), coverage(string), source, type(skip), value(skip)*

*Data is structured according to countries and years of collection.*

*I will include countries, year of survey, sex, number of cars sold, number of people unemployed*

*I Will add primary key of country in all relations.*

# Data Storage Alternatives

*Description: We will study alternatives to storing data in a relational database. Some of the alternatives come from several decades ago, including the hierarchical and network models. Some are newer options, such as NoSQL databases that use JSON or some other encoding. Describe in detail how to store the data using two alternatives to relational databases. Be sure to describe how you would implement the alternatives and the advantages and disadvantages of each.*

*Rubric: Your work will be graded as follows*

* *5 points for clearly describing how your data could be stored using one alternative to relational databases and what the advantages and disadvantages of that approach would be.*
* *5 points for clearly describing how your data could be stored using another alternative to relational databases and what the advantages and disadvantages of that approach would be.*

*Total points possible: 10*

Network Data Model:  
The network database model was created to solve the shortcomings of the hierarchical database model. In this type of model, a child can be linked to multiple parents, a feature that was not supported by the hierarchical data model. The parent nodes are known as owners and the child nodes are called members.

Advantages:  
This model is very simple and easy to design like the hierarchical data model.

In this model, we can access the data easily, and also there is a chance that the application can access the owner’s and the member’s records within a set.

Like a hierarchical model, this model also does not have any database standard.

Disadvantages:

The schema or the structure of this database is very complex in nature as all the records are maintained by the use of pointers.

There’s an existence of operational anomalies as there is a use of pointers for navigation which further leads to complex implementation.

This model does not have any scope of automated query optimization.

NoSQL data model:  
NoSQL essentially describes a grouping of philosophical database design blueprints that avoid relationship data storage. It stores the data as flat files.

Advantages:  
NoSQL is faster then relational databases.

Expansion is incredibly cheap, easy, compared to SQL

It is flexible and changes are easy to make.

Disadvantages:  
It is not yet mature compared to relational data models.

Its speed is largely dependent on computational power of the system.

# Relational Database Design Process

*Description: Consider the list of fields you identified in part c. Identify functional dependencies that exist among them. For each functional dependency, identify the determinants and the fields they determine. This becomes the basis for identifying your entity sets, which then become your tables. Give each entity set or table you identify in this way a unique and clear name, making sure that the names you use are singular nouns. Then list the relationships that exist among the various entity sets. For each relationship, identify its connectivity (one-to-one, one-to-many, many-to-many) and participation (optional or mandatory). Finally, make sure that none of the attributes you've assigned to each entity set are multi-valued. If they are, take the steps needed to break them down.*

*Rubric: Your work will be graded as follows:*

* *8 points for identifying all the functional dependencies, including determinants and the columns whose values they determine.*
* *2 points for naming the entity sets that make up your data with clear, easy-to-understand names.*
* *6 points for identify the relationships among the entity sets and identifying connectivity and participation for each.*
* *2 points for breaking down multi-valued attributes.*

*Total points possible: 18*

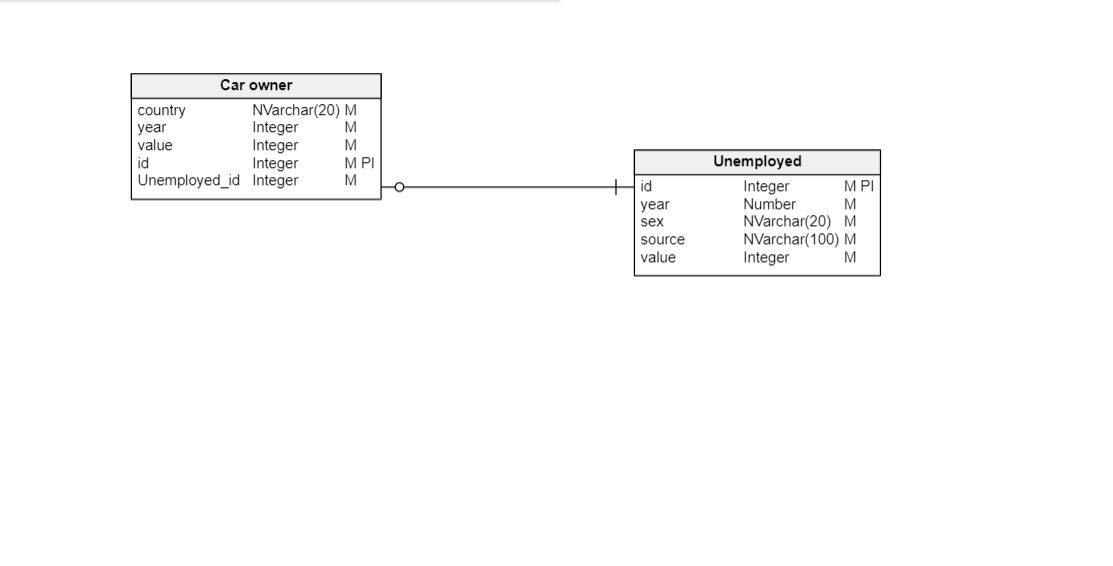
Country,Year->Unit,Value

Country, Year, Sex, Source->Value

Names of Entities:  
Country

Unemployed

Car owner



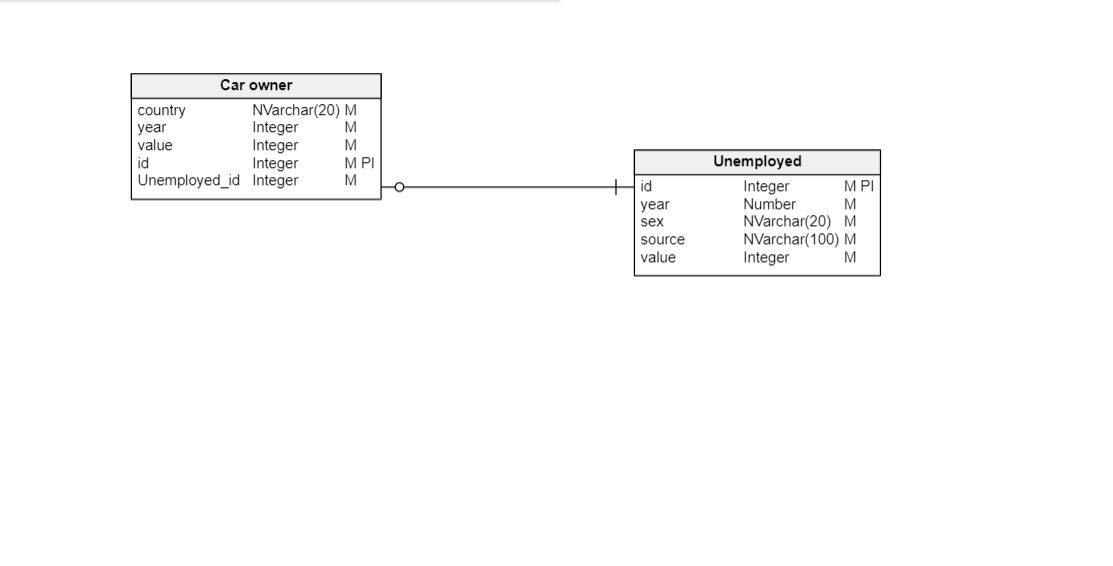
# Relational Database Design

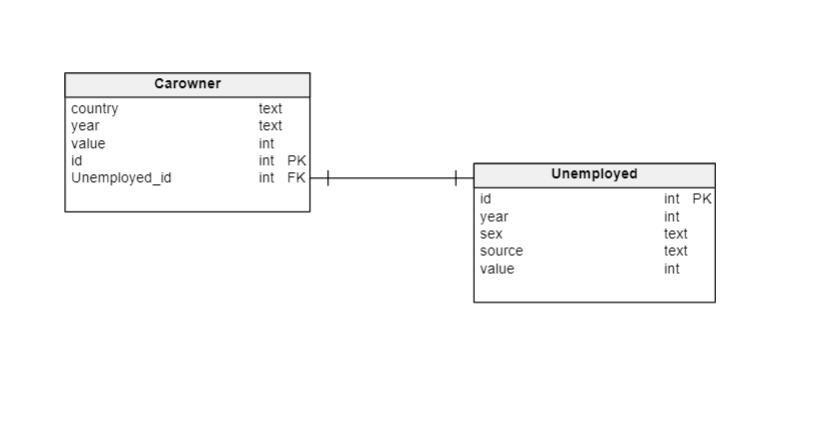
*Description: This is where you will complete your database design. For each of the entity sets you identified in the preceding section, analyze them to make sure they pass 2nd, 3rd, 4th, and Boyce-Codd Normal Form. If they do not, introduce additional entity sets or key changes to make sure that they do. Then, add foreign keys to connect entity sets that are related. For many-to-many relationships, introduce bridge entity sets to convert them into two one-to-many relationships. Also, consider whether you should introduce surrogate keys to create a more efficient primary key for some of your entity sets. Finally, diagram your design in Vertabello. Make sure your ER diagram correctly shows all entity sets, their primary and foreign keys, the data types for each attribute, and the connectivity and participation characteristics of each entity set. Your final Vertabello design should be something you could actually implement in a relational database management system.*

*Rubric: Your work will be graded as follows:*

* *4 points for the normalization analysis of your entity sets.*
* *3 points for introducing bridge entity sets.*
* *3 points for choosing foreign keys and perhaps more efficient surrogate keys*
* *10 points for correctly depicting your physical database model in Vertabello*

*Total points possible: 20*

4th normal form  


Physical model:  


# Data Definition Language (DDL) Scripts

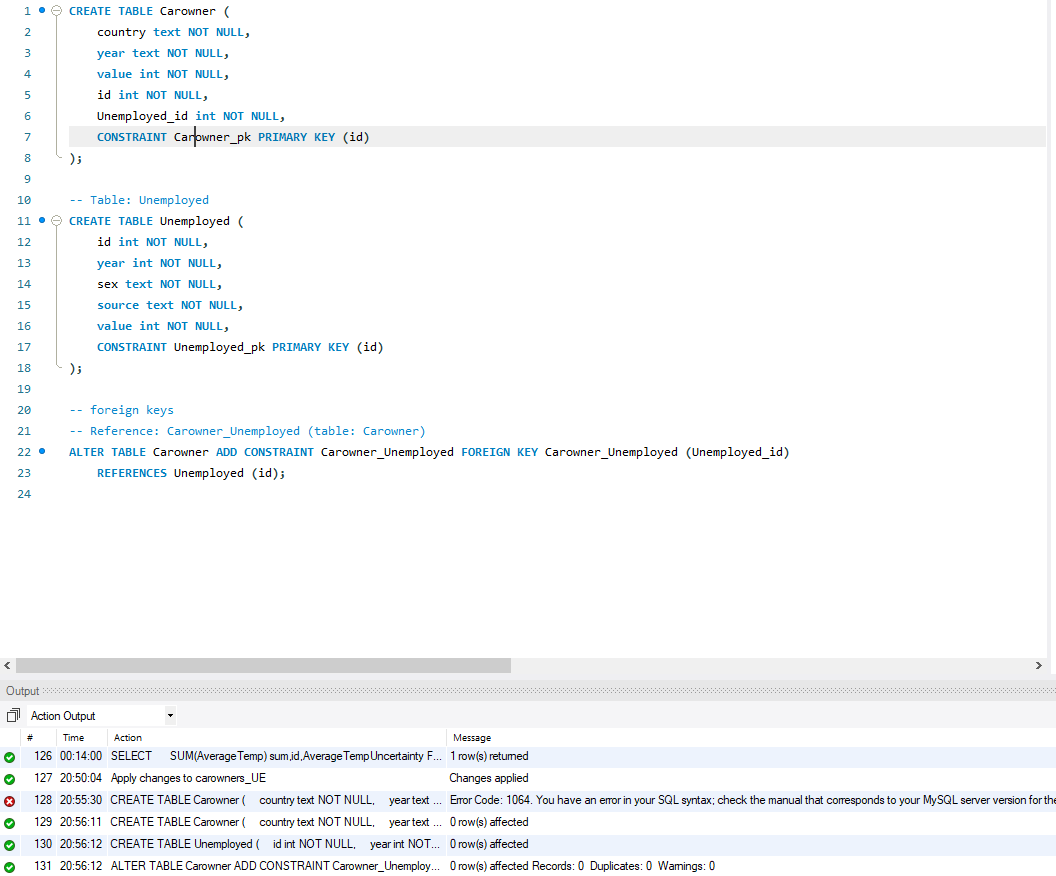
*Description: Use Vertabello to generate a script of SQL commands that build the database and its table structures. Write scripts or build Excel spreadsheets that take your data files and generate scripts of SQL insert statements from them. Use the MySQL source command to run the various scripts needed to build and populate the database in MySQL. Include the source code and / or Excel spreadsheets you use to manipulate and populate the data. Make sure all your tables have at least three records in them and that you've linked the tables through their foreign keys.*

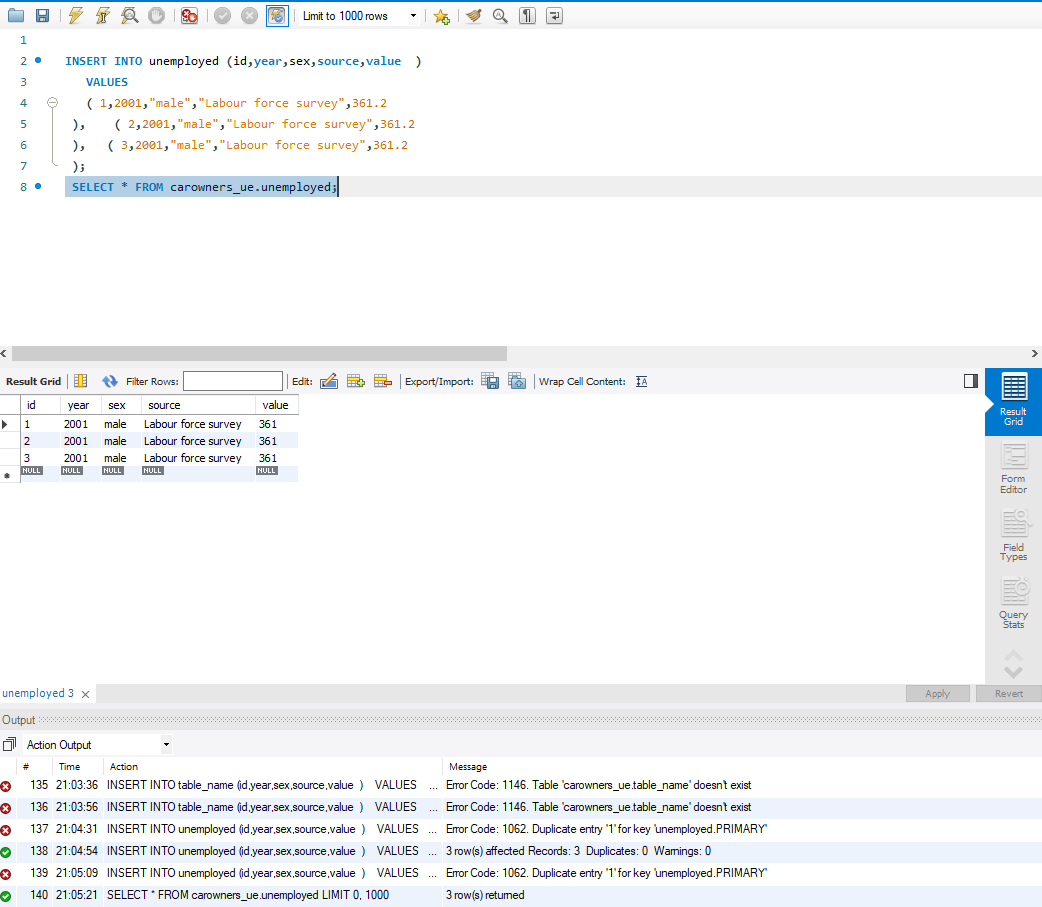
*Rubric: Your work will be grades as follows:*

* *Database and table creation statements from Vertabello saved as an sql script file: 3 points*
* *Scripts you write or Excel spreadsheets you create to generate SQL commands for populating the tables, uploaded to GitHub: 8 points*
* *Descriptions of the scripts and Excel spreadsheets you wrote along with code excerpts included in the Word document: 5 points*
* *Screenshots of your successful attempts to use the MySQL source command to populate each table with at least three records: 4 points*

*Total points possible: 20*

The script created are used to create tables in database and also define relationship among them.





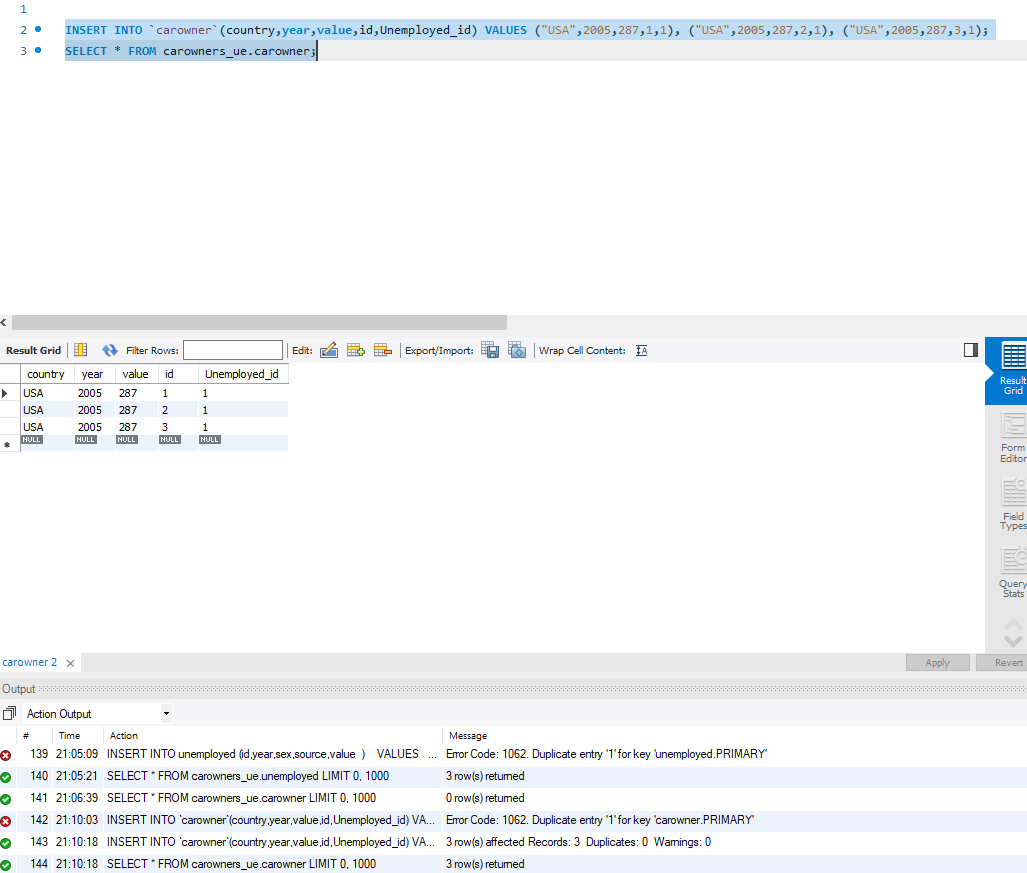
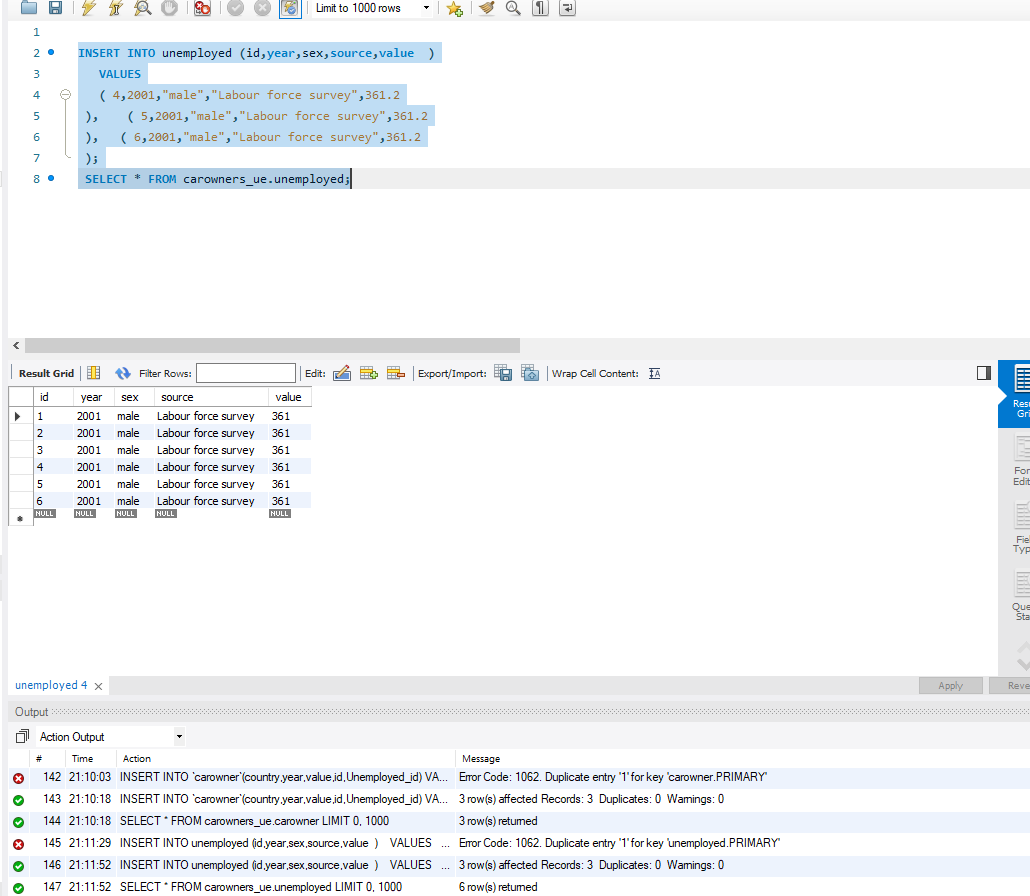
# Data Manipulation Language Scripts

*Description: Write the SQL commands for twelve queries. Two queries should be insert statements, two should update statements, one should be a delete statement, one should be a simple select statement that selects a subset of the rows and columns from one table, two should be a select statements that select data from a joining of two tables, two should use summary functions to generate statistics about the data, one should be a multi-table query, and one should be another query of your choice. Show the queries and screenshots of the results in your Word document, and save your queries in a commented sql script to GitHub.*

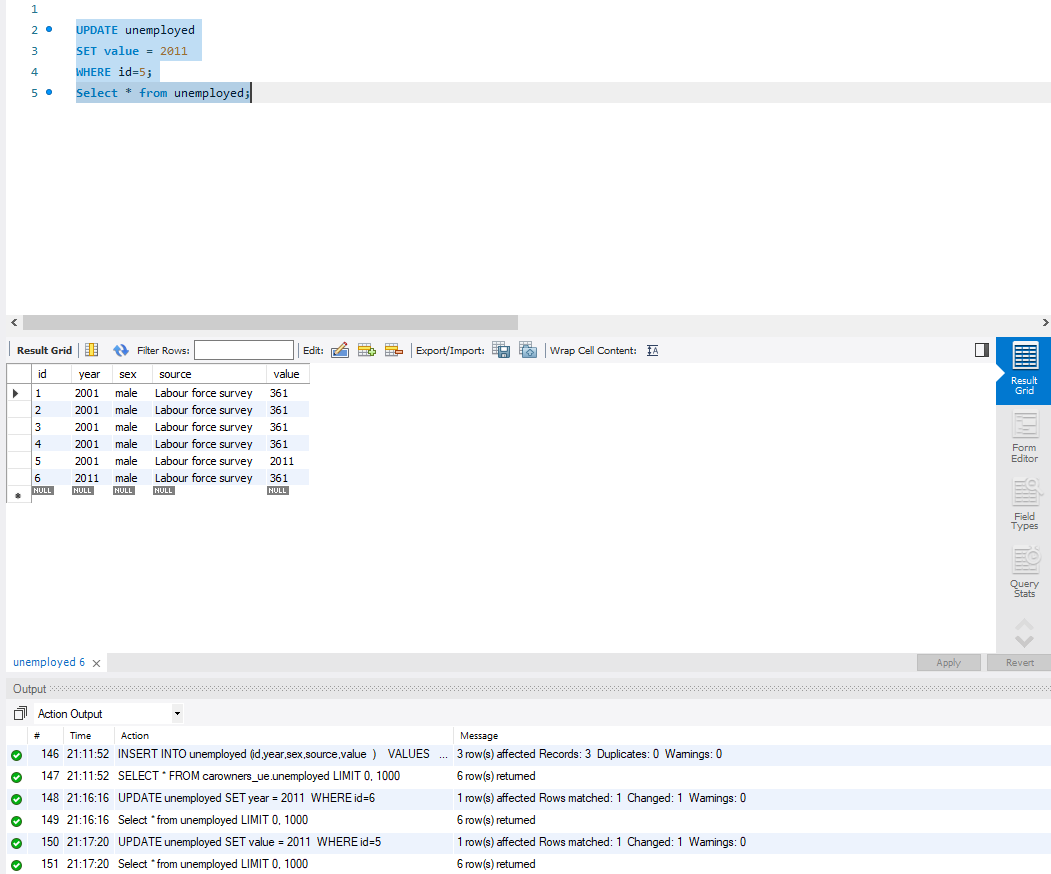
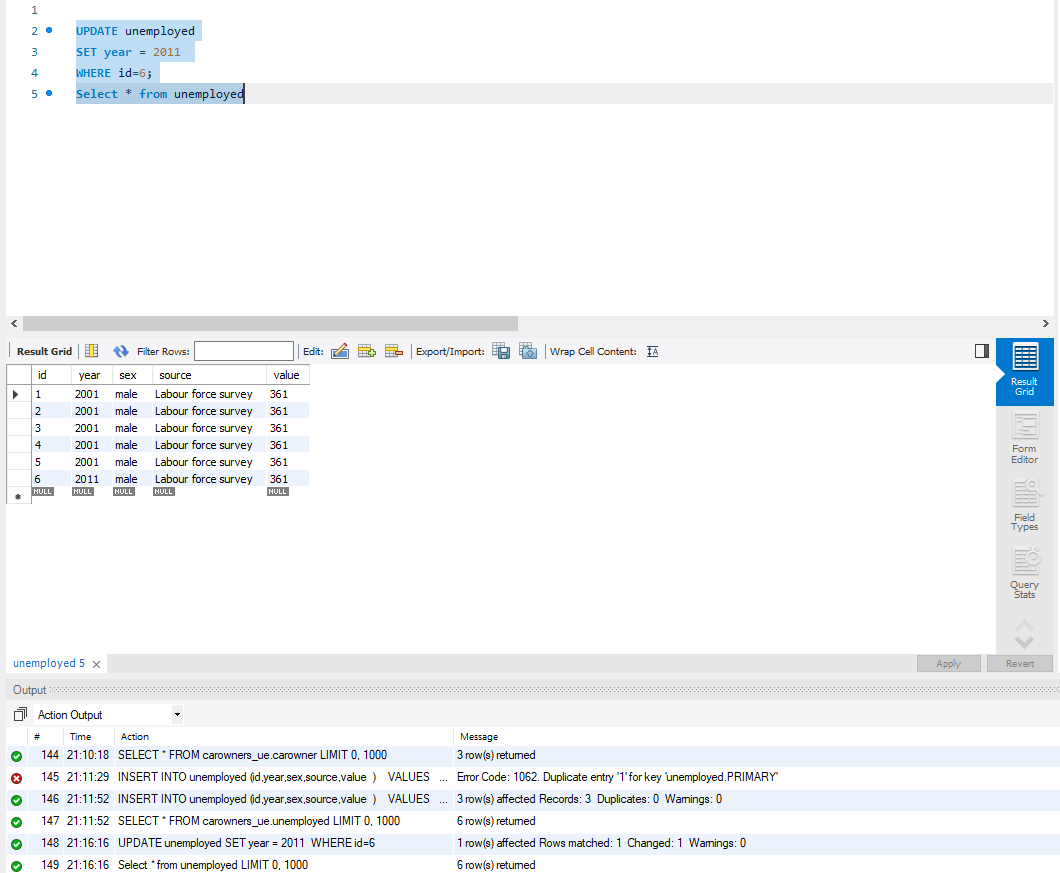
*Rubric: Your work will be graded as follows:*

* *1 point each for the two insert statements*
* *1 point each for the two update statements*
* *1 point for the delete statement*
* *1 point for the simple select statement*
* *2 points each for the 2 join statements*
* *2 points each for the two that use summary statements*
* *2 points for the multi-table query*
* *2 points for the query of your choice.*
* *12 points for showing the query and a screenshot of the corresponding result set back-to-back for each of these queries in your Word document.*

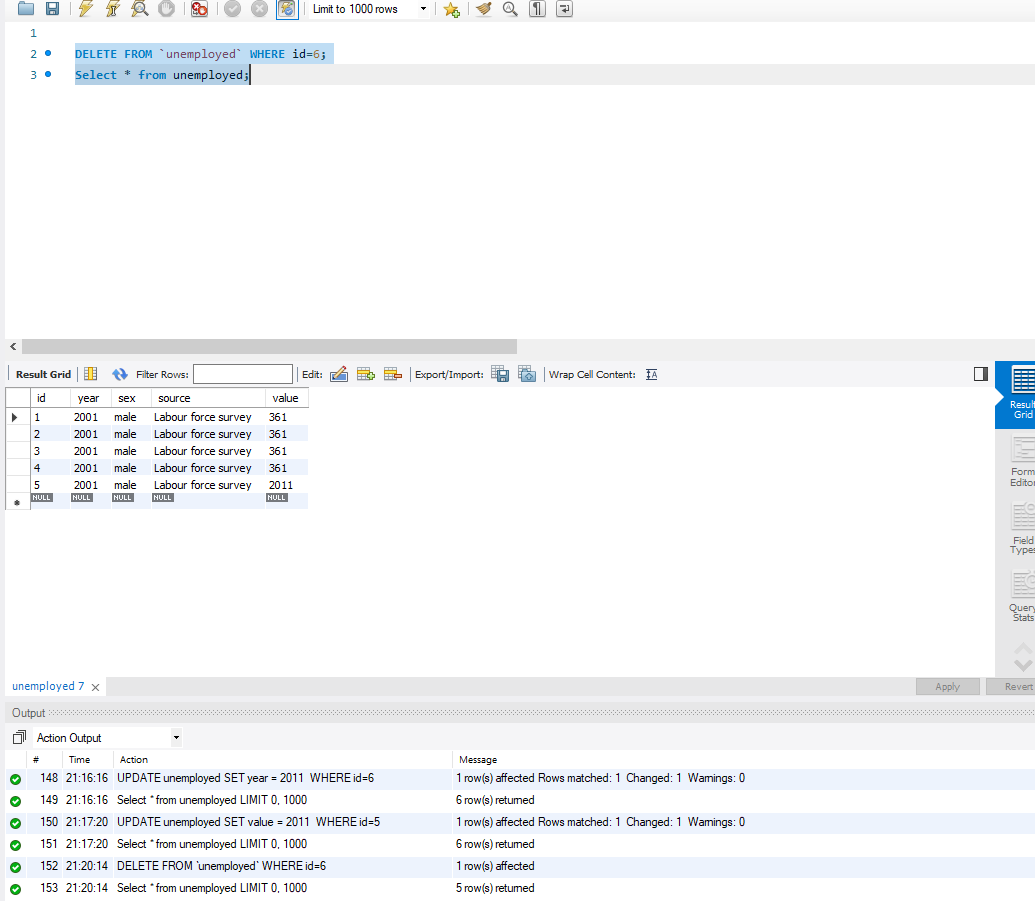
*Total points possible: 30*

INSERT COMMANDS  


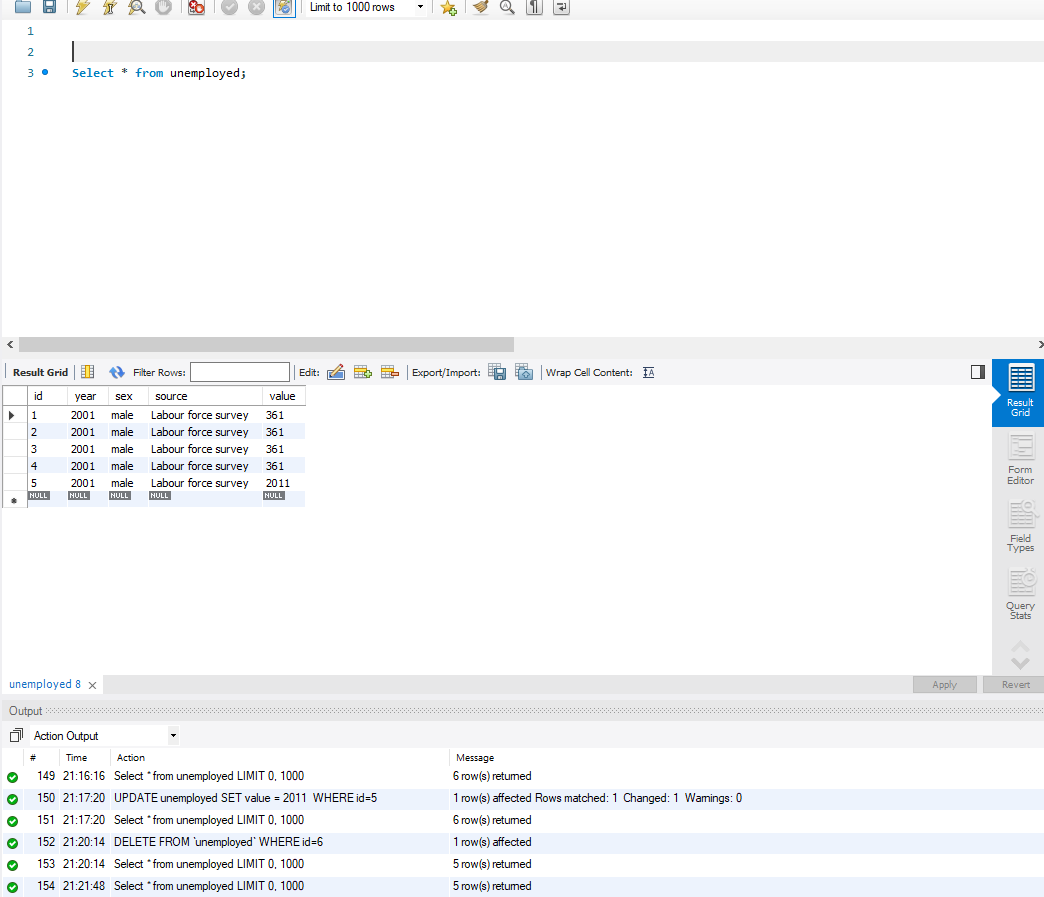
UPDATE COMMANDS:

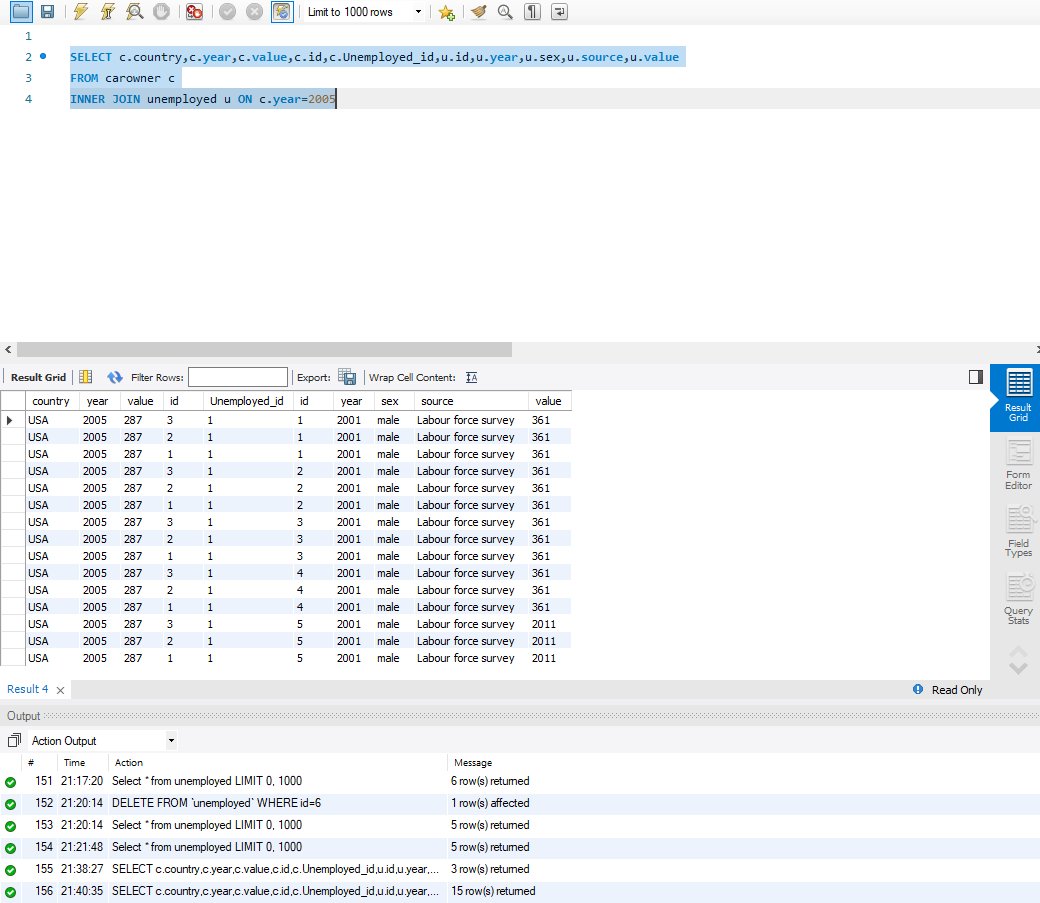
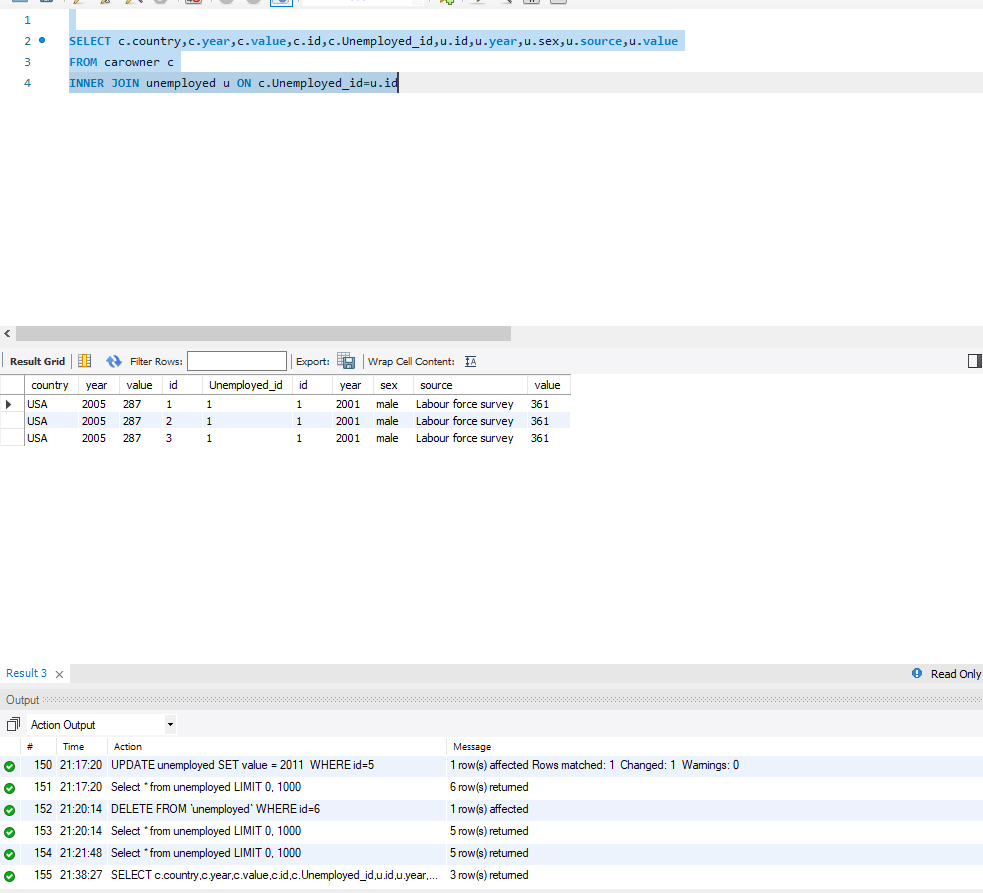


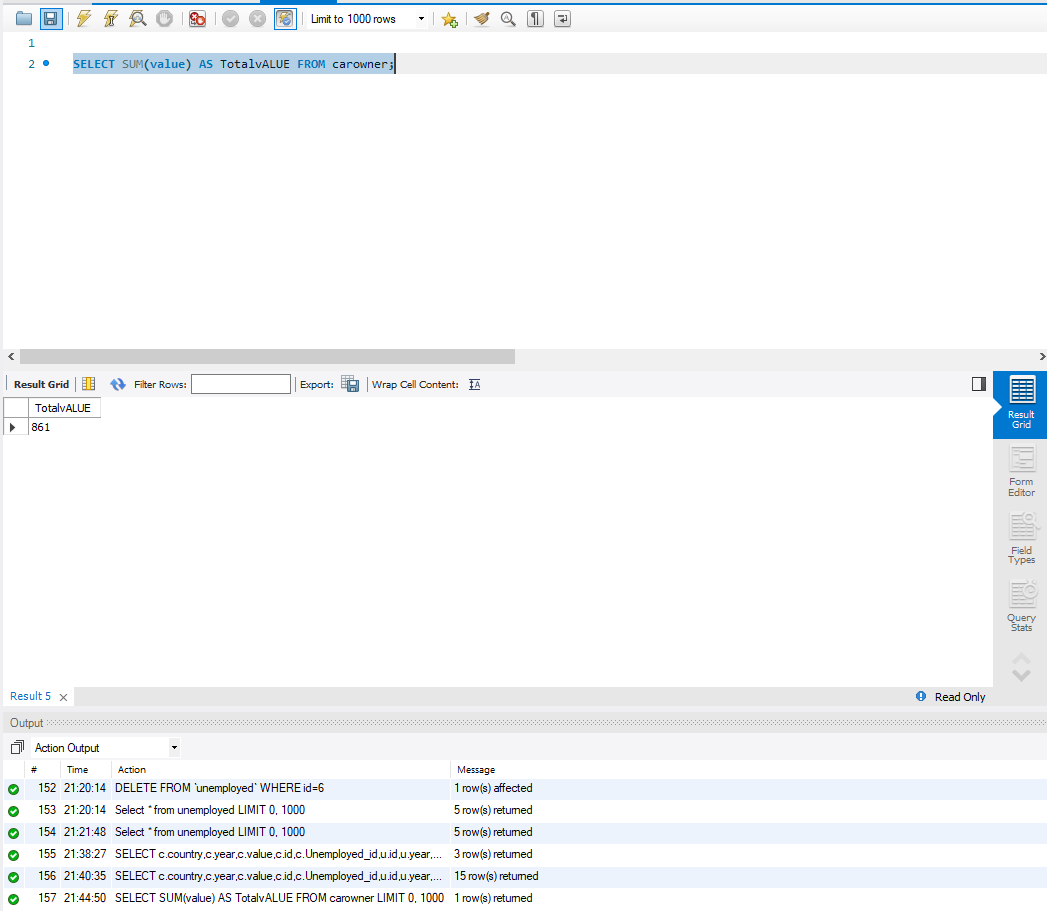
DELETE COMMAND:



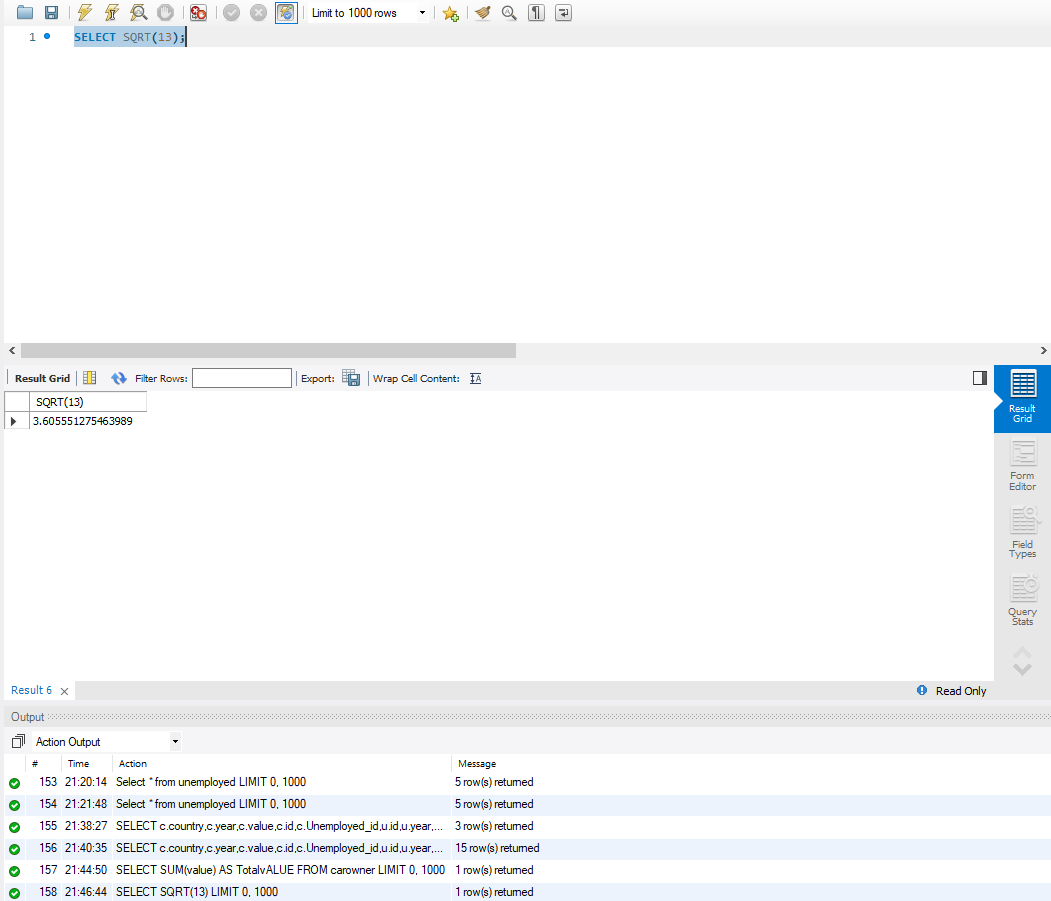
SELECT COMMAND:



JOIN COMMANDS:  


SUMMARY COMMAND:  


COMMAND OF CHOICE:



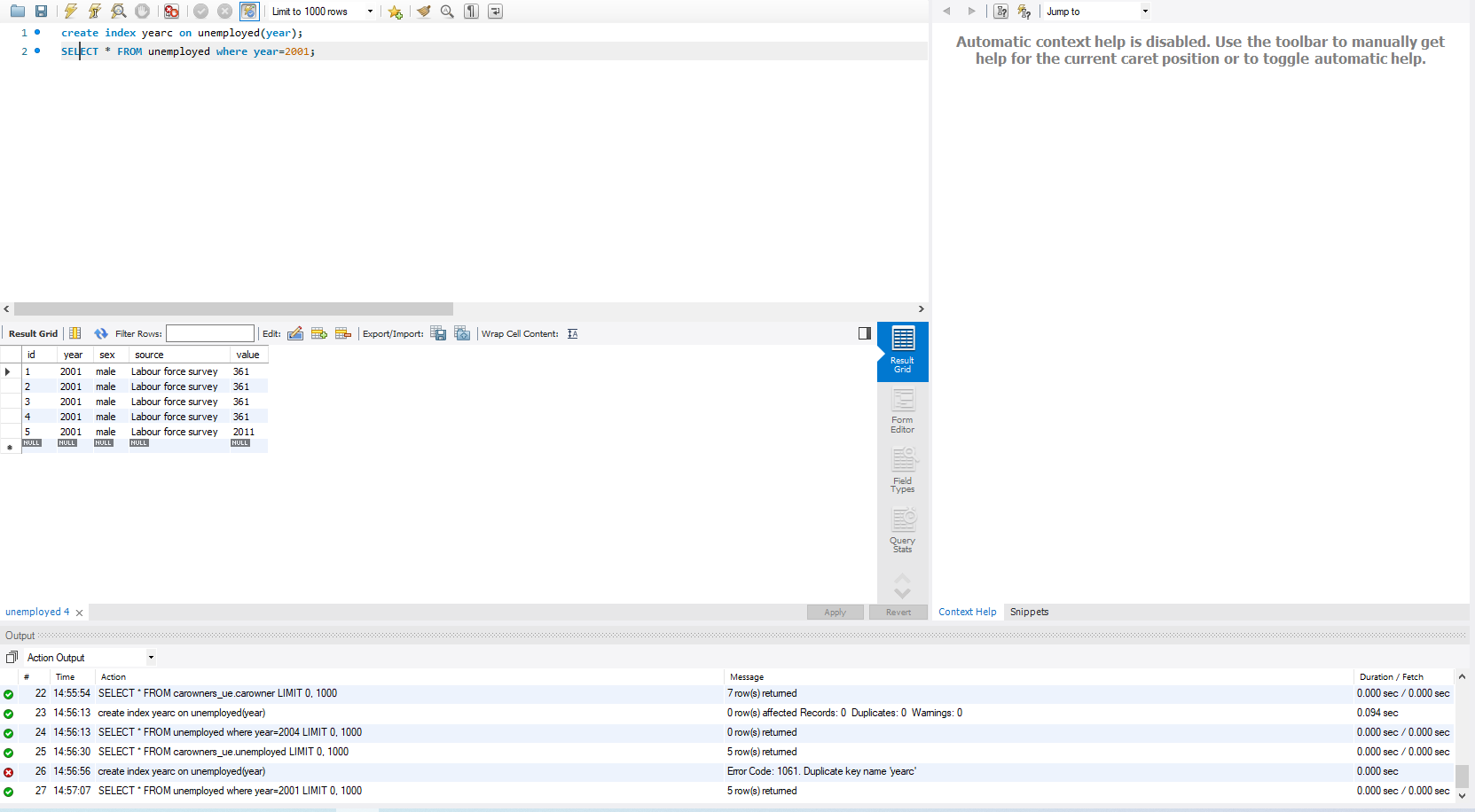
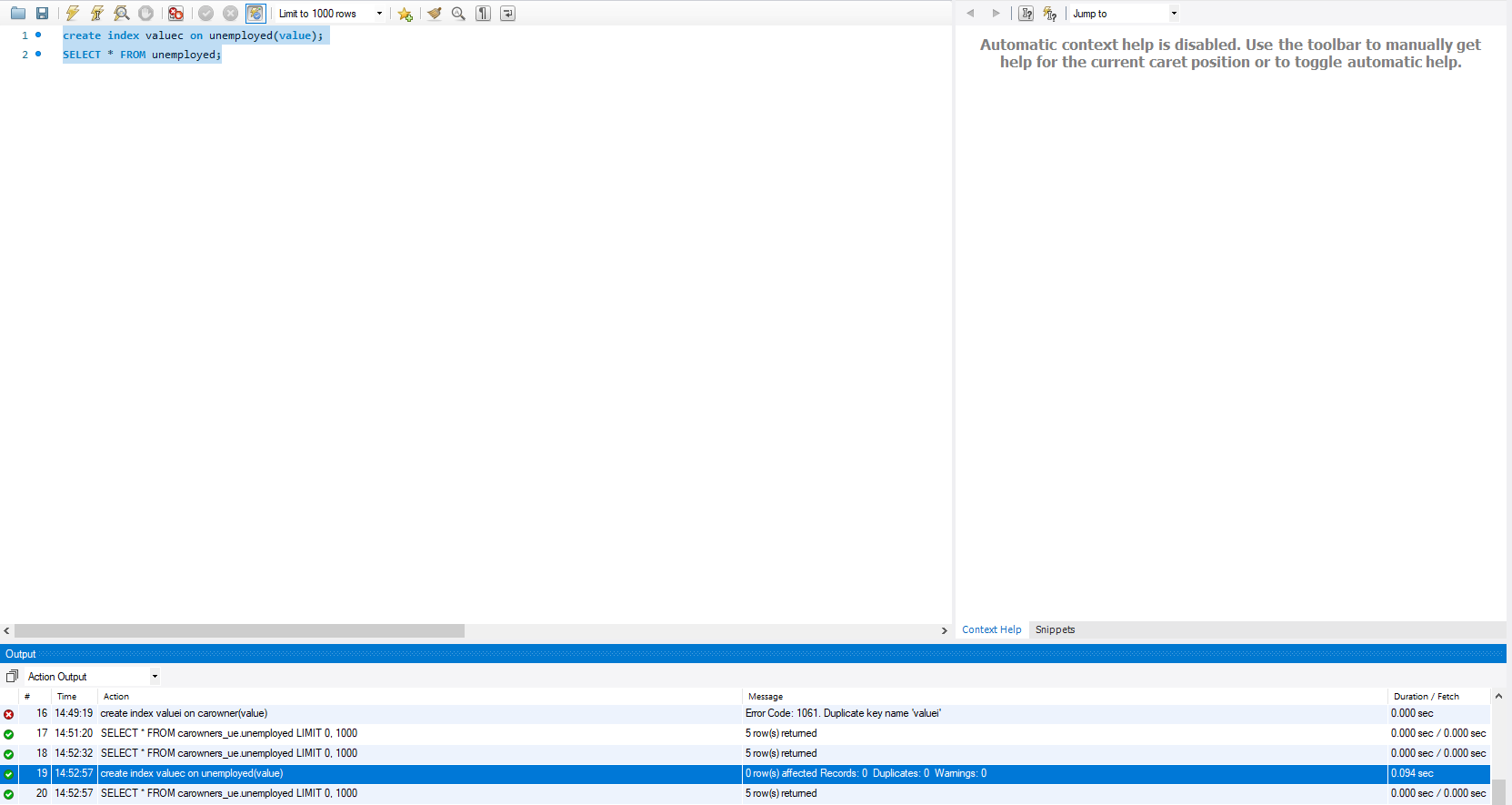
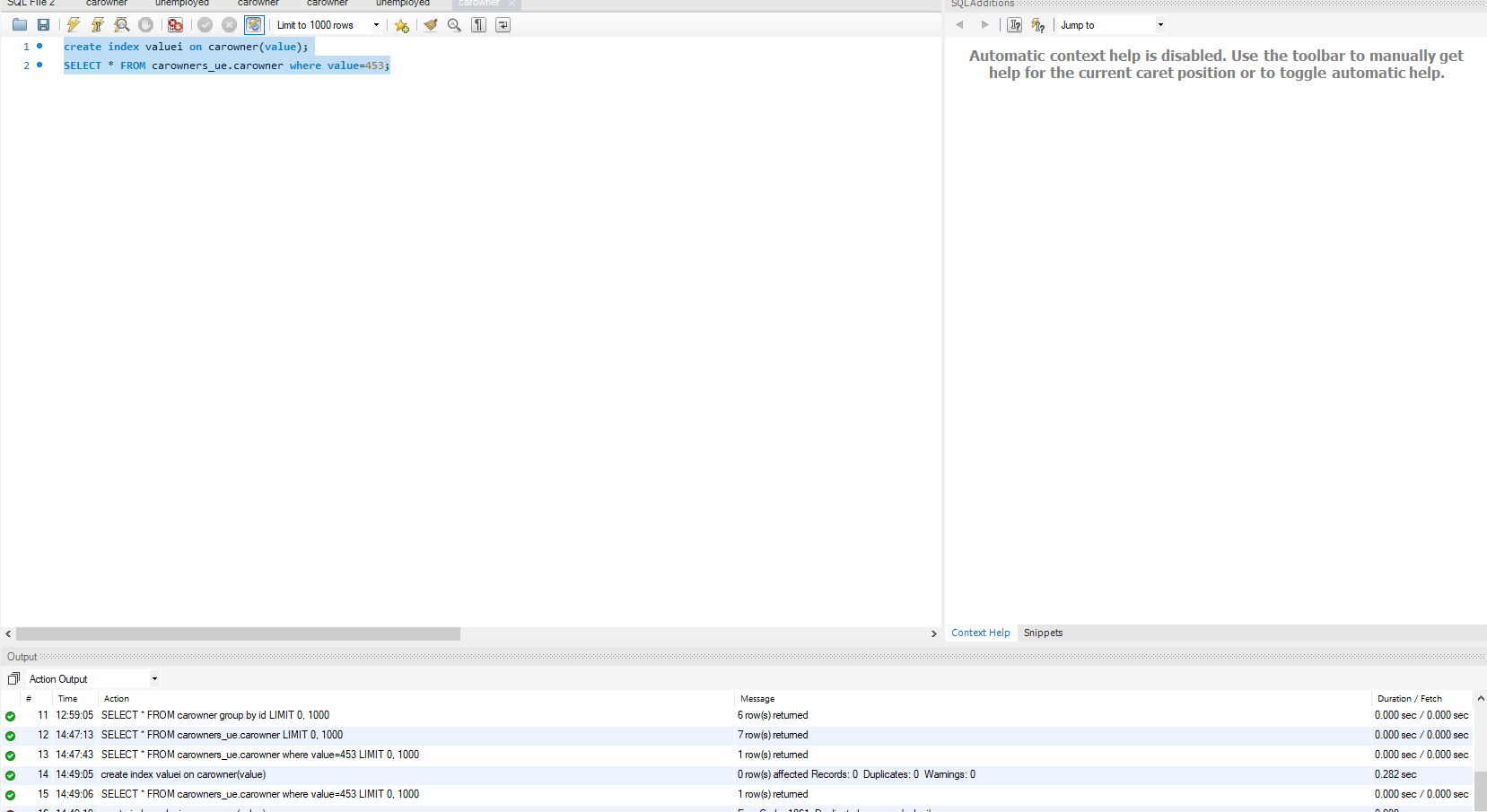
# Indexes

*Description: Improve the performance of your design by adding indexes to various tables. Show the SQL needed to add the indexes. Explain why you chose the ones you added. Explain how you would demonstrate the impact the indexes had on the performance of various queries.*

*Rubric: Your work will be graded as follows:*

* *6 points for clearly defining at least three indexes and explaining why you chose them.*
* *3 points for showing the sql needed to generate the indexes*
* *3 points for explaining how you would demonstrate the performance improvement afforded by the indexes.*

*Total points possible: 12*



I have created 3 indexes on two tables. The performance improvement will show when the data in these tables will grow. As of now time taken for query is 0.00 seconds.

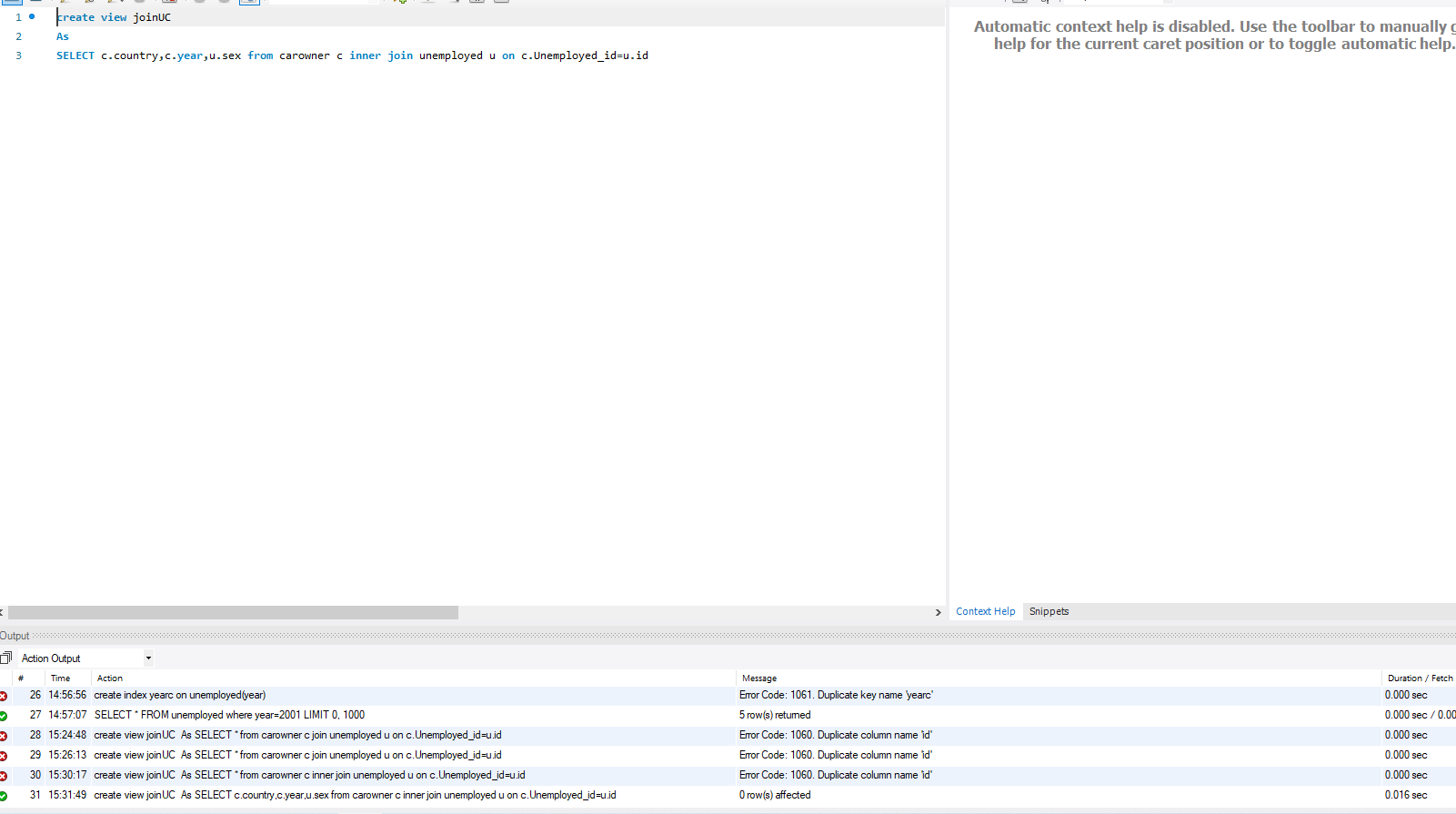
# Views

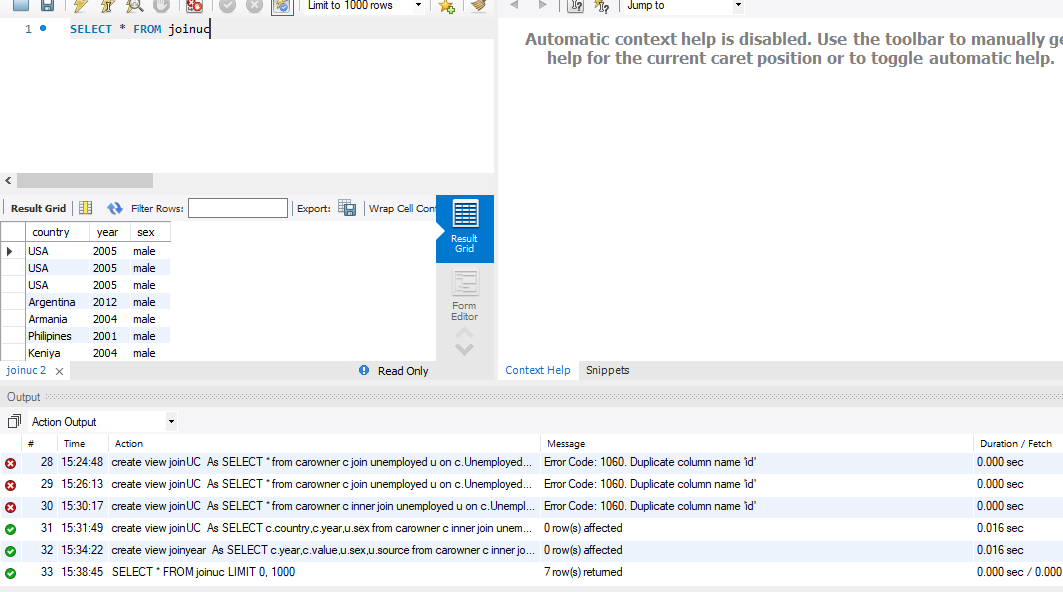
*Description: Add two views to your database to provide easy access to combinations of data from multiple tables.*

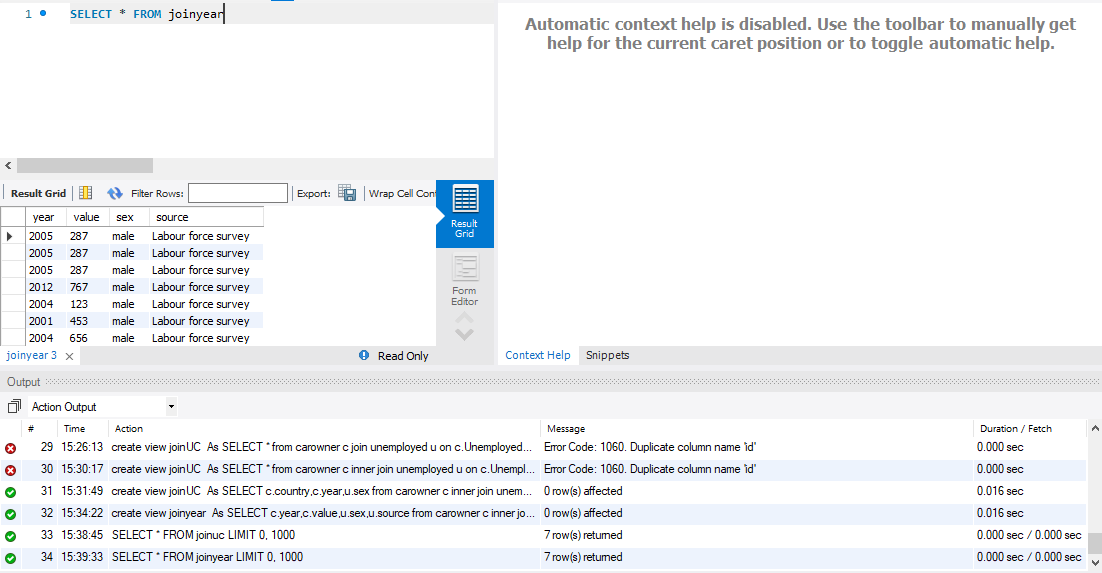
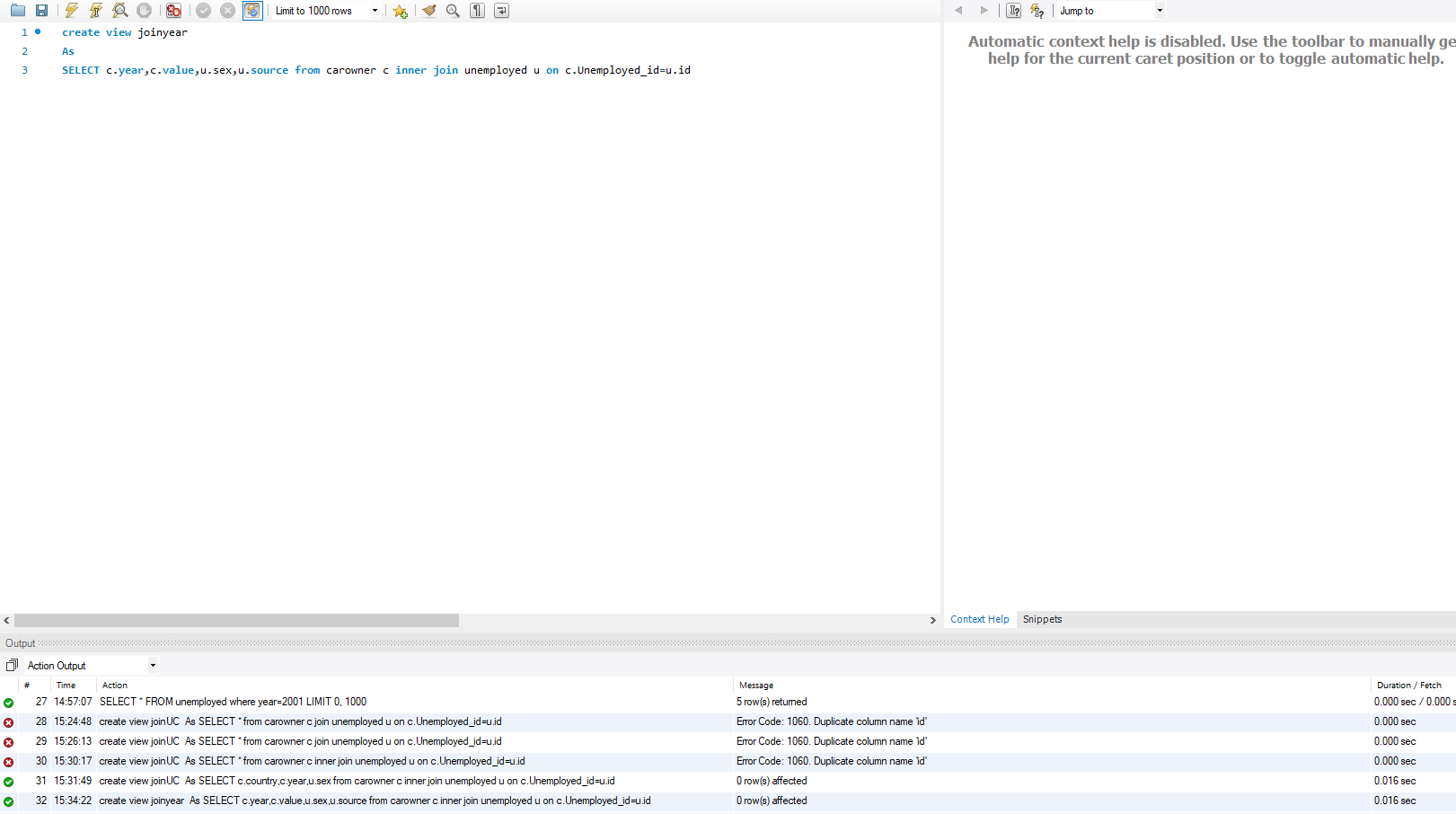
*Rubric: Your work will be graded as follows:*

* *2 points for including the SQL for generating the two views in your Word document*
* *2 points for including screenshots for the data contained in each view in your Word document*
* *2 points for explaining why each view is a valuable addition to your database*
* *2 points for explaining who might benefit most from having access to each view.*

*Total points possible: 8*







These 2 views are used to store valuable queries for later used getting data

# Triggers

*Description: Add a trigger to a table so that data will be updated when a certain event occurs*

*Rubric: Your work will be graded as follows:*

* *2 points for including the SQL for the trigger in your Word document*
* *2 points for clearly explaining the purpose of the trigger*
* *2 points for a screenshot and explanation that shows the trigger in action.*

*Total points possible: 6*

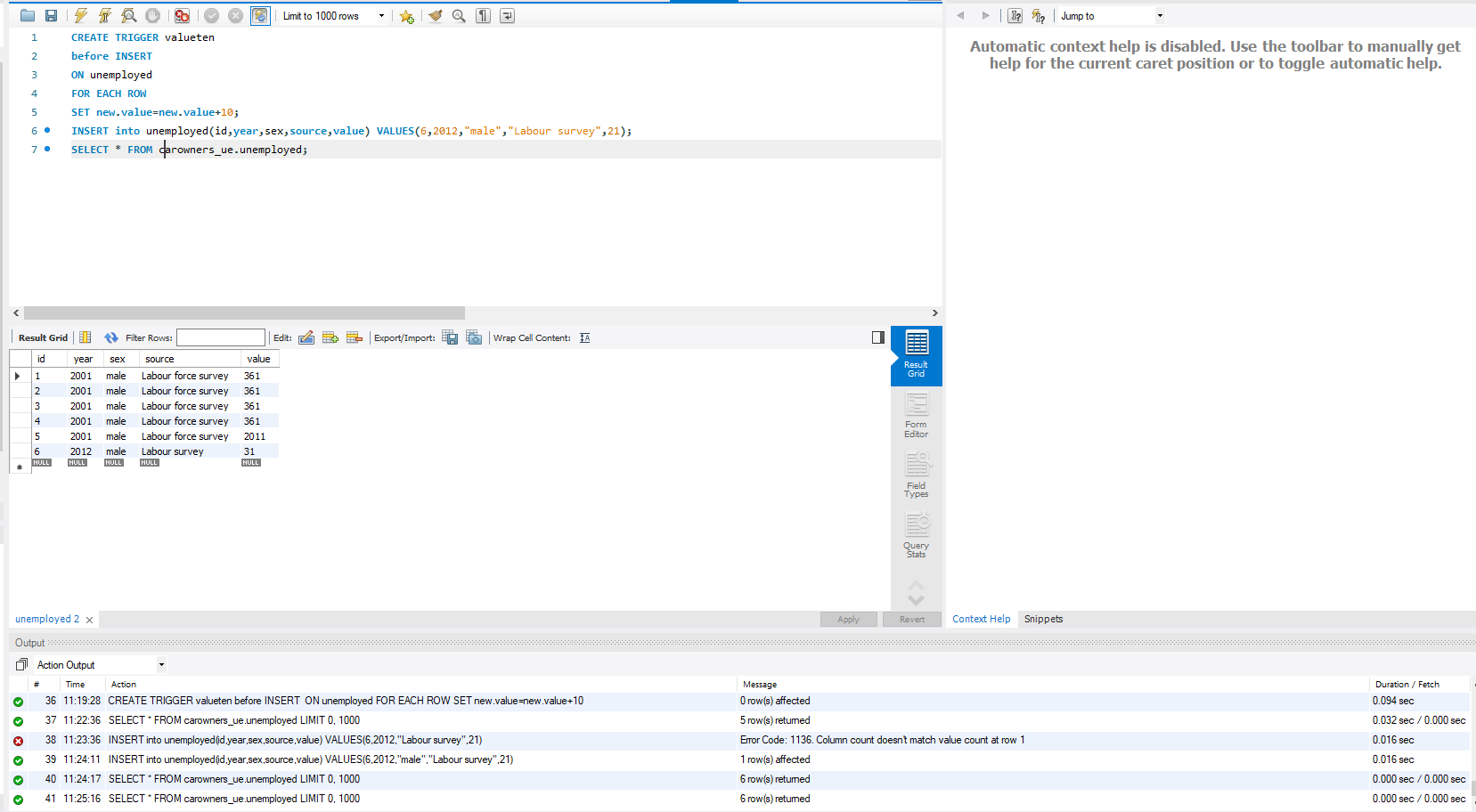
CREATE TRIGGER valueten

before INSERT

ON unemployed

FOR EACH ROW

SET new.value=new.value+10;

Explanation:  
This trigger will add a value of 10 to whatever is being inserted in each row  
 

# Transactions

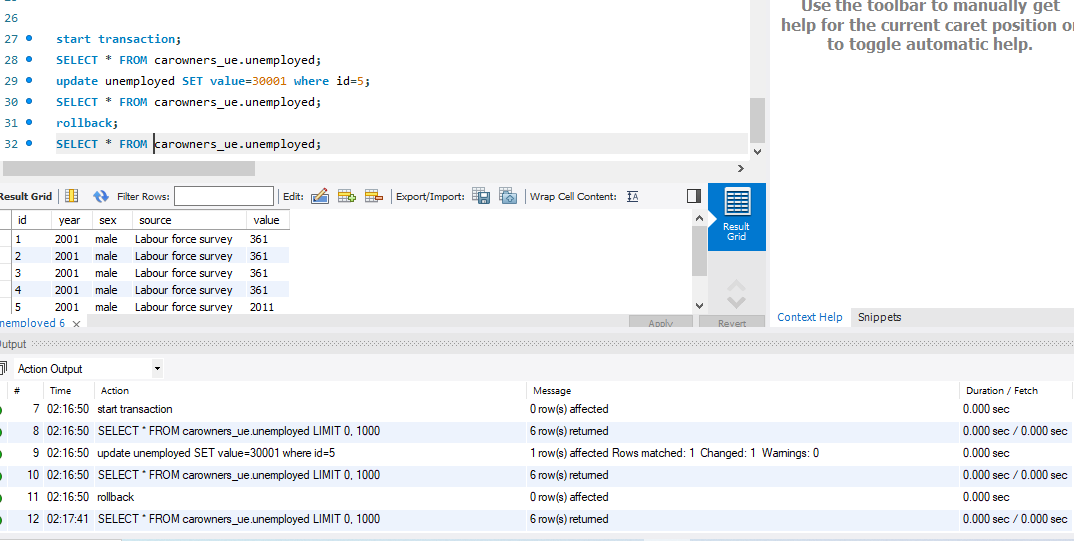
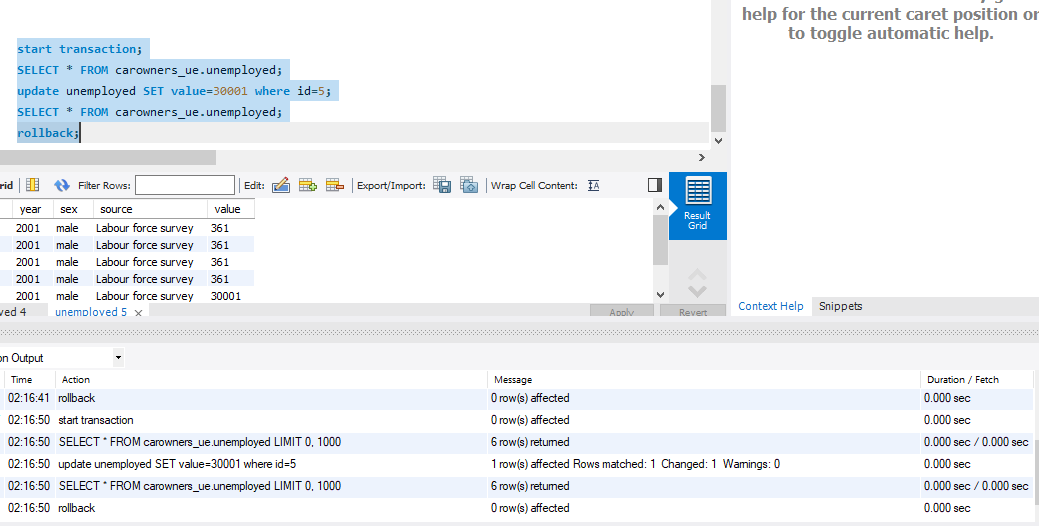
*Description: Demonstrate that you know how to define and use a transaction. Why are transactions important for ensuring ACID behavior?*

*Rubric: Your work will be graded as follows:*

* *3 points for clearly explaining the importance of transactions to ensuring ACID behavior*
* *3 points for including a screenshot and accompanying explanation of a MySQL transaction.*

*Total points possible: 6*

Atomicity in ACID means the transaction exhibit completeness of process. Consistency for correctness, Isolation means that transactions can run at the same time, and durability means that a transaction will not impact the state of data if transaction ends abnormally.



We used a transaction to demonstrate the concept of ACID where until a transaction is committed we can rollback at any time.

# Database Security

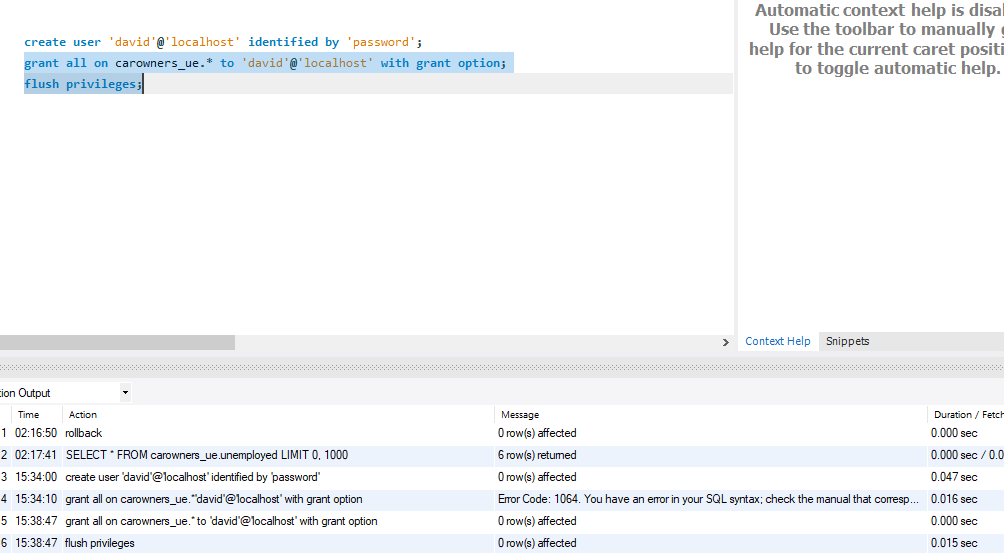
*Description: Identify the different kinds of users who will use your database. Write GRANT statements to define the privileges for these different kinds of users.*

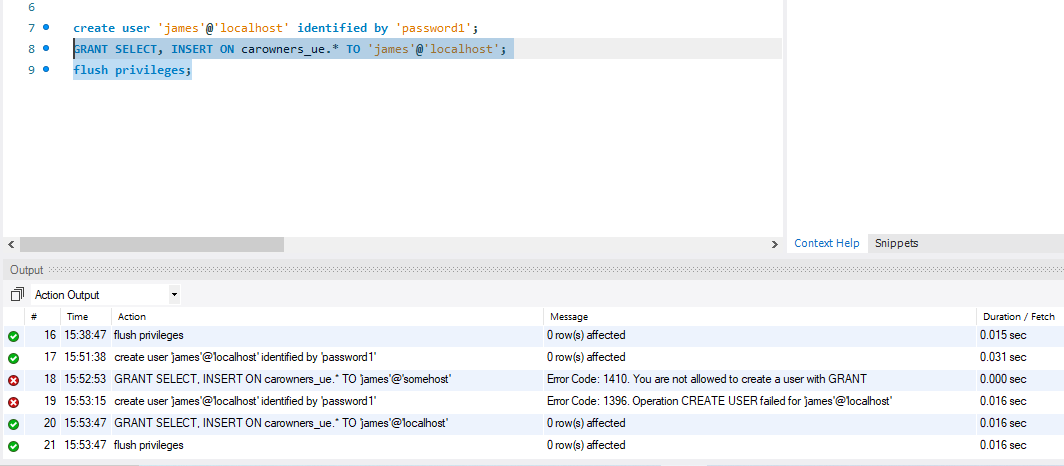
*Rubric: Your work will be graded as follows:*

* *6 points for clearly identifying and describing the various kinds of users who will use the databases and identifying and justifying what privileges each should have.*
* *4 points for writing GRANT statements that assign privileges to these different kinds of users.*
* *4 points for demonstrating with screenshots that your GRANT statements do distinguish among different kinds of users in regard to what they can do with the database.*

*Total points possible: 14*

David will use the database with all the privileges of a super user including grant.



James has privileges of select and insert on all tables in the database  


# Locking and Concurrent Access

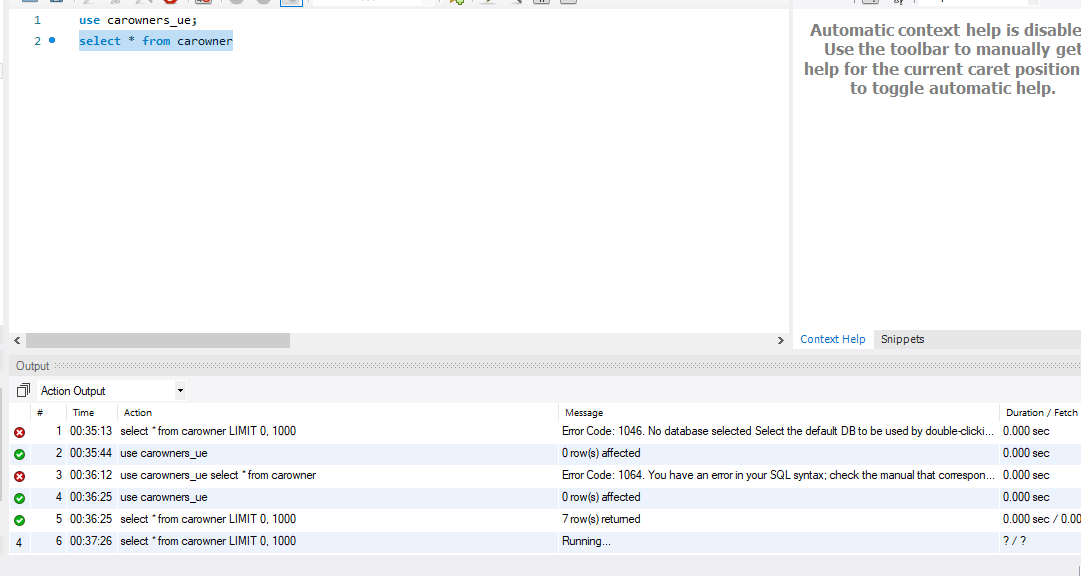
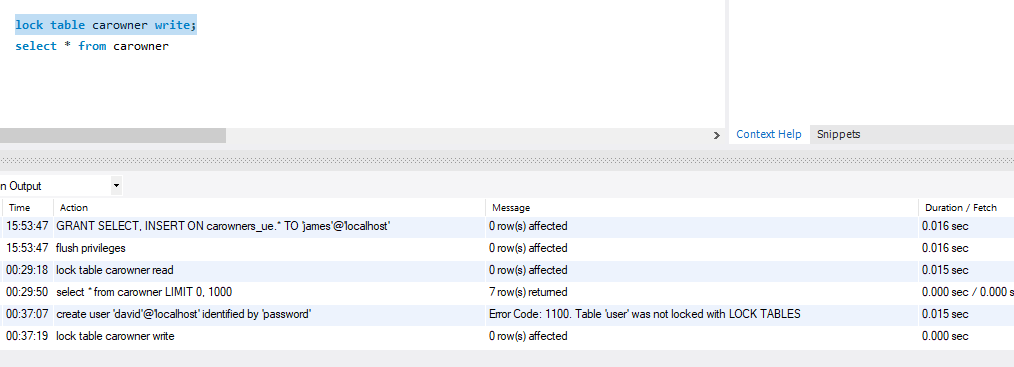
*Description: Explain the purpose of locking tables and show how to do that to prevent inconsistencies that may arise in your data when concurrent transactions take place.*

*Rubric: Your work will be graded as follows:*

* *3 points for clearly explaining an example that shows why you should lock tables to prevent inconsistencies.*
* *3 points for providing a screenshot and accompanying explanation of locking tables.*

*Total points possible: 6*

**MySQL allows a client session to acquire a table lock explicitly to cooperate with other sessions to access the table's data**.

**Locking table write and resultantly other user can’t select anything from locked table  
**

# Backing Up Your Database

*Description: How you will back up your database. What commands will you issue? How frequently will the commands run? How can they be automated? Where will the backups be stored?*

*Rubric: Your work will be graded as follows:*

* *12 points for clearly explaining and justifying your database backup strategy, including the frequency with which you will back up the database, how you will automate backups, where you will store them, and how you will secure them. You will earn three points for addressing each factor (frequency, location, automation, and security)*
* *3 points for providing a screenshot of the command you would issue to back up the database and for including a portion of the resulting file.*

*Total points possible: 15*

mysqldump -h *localhost* -u *david* -p password *carowners\_ue* > *carowners.sql*

# Python Programming

*Description: Write a Python program that generates a report that contains a subset of the data from your database. Include the code for your Python program in your Word document, and also post the program to your GitHub repository.*

*Rubric: Your work will be graded as follows:*

* *12 points for writing a Python script (and including its code in the Word doc) that will pull data from a database and store it to a text file and present it to the screen. Your code must have comments in it that explain how it works. You will be awarded 3 points for successfully connecting to the database, 3 points for successfully querying it, and 4 points for presenting the data to the screen and to a file. Internal comments count for 2 points.*
* *2 points for posting the code to GitHub*
* *4 points for showing a screenshot of your running the script and showing the results it produces on the screen.*

*Total points possible: 18*

#connector to connect with mysql from python

import mysql.connector

from mysql.connector import errorcode

#a function which will be called to connect with db and execute a query

def select(conn,query):

#a cursor refers to the point of execution

cursor = conn.cursor()

cursor.execute(query)

#list declaration

results = []

#fetches all the records

for row in cursor.fetchall():

results.append(row)

cursor.close()

return results

def execute(conn,query): # update, delete, and insert

cursor = conn.cursor()

cursor.execute(query)

conn.commit()

def show(rows):

for row in rows:

print(row)

#connection attempt

try:

conn = mysql.connector.connect(

user="root",

password="",

host="localhost",

database="carowner\_ue")

except mysql.connector.Error as err:

print("Cannot connect.")

exit()

rows = select(conn,"select \* from carowner c join unemployed u where c.id=u.id")

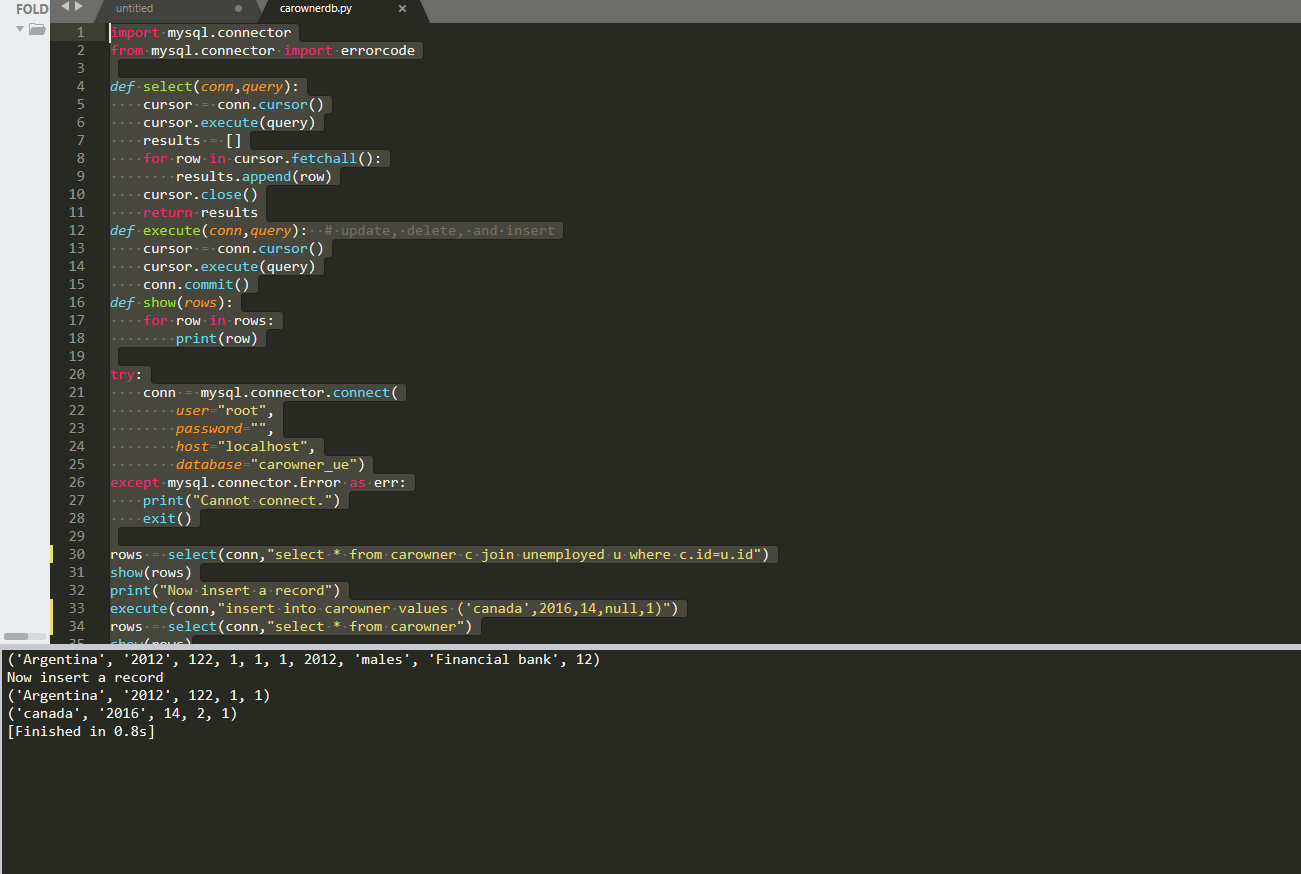
show(rows)

print("Now insert a record")

execute(conn,"insert into carowner values ('canada',2016,14,null,1)")

rows = select(conn,"select \* from carowner")

show(rows)



# PHP Programming

*Description: Build an HTML form that enables the user to specify criteria to search by. Use PHP to show the results of the query on a resulting web page. Make sure you include protections against an SQL injection attack. Include your HTML and PHP code in your Word document, and also post the files to your GitHub repository.*

*Rubric: Your work will be graded as follows:*

* *4 points for writing an HTML form the user will use to enter search criteria*
* *8 points for a PHP script that uses the search criteria and returns results*
* *4 points for an HTML page that shows the results*
* *4 points for explaining what SQL injection might be run on your website and explaining how you prevented it.*
* *4 points for providing screen shots of your PHP website in action.*
* *2 points for posting your code to GitHub*

*Total points possible: 26*

Search form and script that returns search results

<form action="search.php" method="POST">

    <input type="text" name="search" placeholder="search" >

    <button type="submit" name="submit">search</button>

</form>

<?php

include\_once("config.php");

include\_once("Database.php");

$db = new Database();

$db->query("select \* from carowner");

$db->query("select \* from carowner;");

//$db->bind(':country', "canada"); //'"Ray" or 1=1; --');

$results = $db->resultSet();

echo "enter id to search from database : \n";

foreach ($results as $result) {

    echo "<table>"."<tr>"."<td>".$result->country. "\t" . $result->year . "\t" .

       $result->value . "\t" . $result->id. "\t"."</td>" ."</tr>"."</table>"."\n";

}

$db->query("insert into carowner values ('newzealand',2014,2334,null,1)");

$db->execute();

?>

This shows the search result page and script that translates the search result

include\_once("config.php");

include\_once("Database.php");

$db = new Database();

?>

<h1>search page</h1>

<div>

<?php

if(isset($\_POST['submit'])){

$a= $\_POST['search'];

$db->query("SELECT \* FROM carowner where

id LIKE '%$a%'

");

$result = $db->single();

// foreach ($results as $result) {

    echo $result->country . "\t" . $result->year . "\t" .

       $result->value . "\t" . $result->id ."\n";

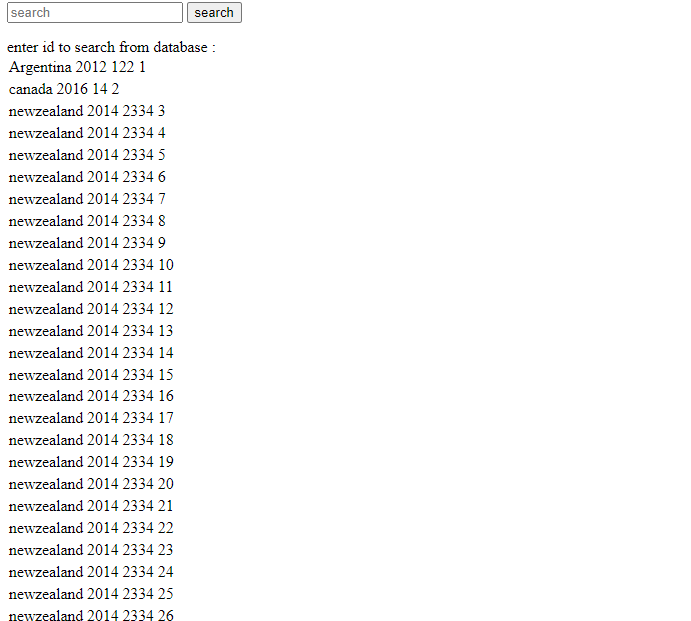
// }

}

?>

</div>

Screenshots that shows the search page and search results



The PDO method is used to prevent SQL injection to happen on any application

# Suggested Future Work

*Description: Describe the limitations of your current database and explain how you or someone else could improve the design to address these shortcomings. Also describe how you might take advantage of leverage cloud services to increase the performance and availability of your database. Finally, explain the advantages and disadvantages of storing your data in a NoSQL format instead.*

*Rubric: Your work will be graded as follows:*

* *3 points for clearly describing the limitations of your databases*
* *3 points for explaining how you would address these shortcomings*
* *3 points for explaining how you might migrate the database to the cloud and describing what advantages you might gain from doing that.*
* *3 points for explaining the advantages and disadvantages of storing your data in a document-based NoSQL format instead.*

*Total points possible: 12*

Our database does not yet show patterns.  
It is limited in terms of data storage capacity

We would implement Big data

# Activity Log

*Description: As an appendix, the team will keep a daily diary or log of their activity. What did you or your team study in this class each day? What did you learn? What did you accomplish or build or design? You don't have to enter something every day, but there should be at least three entries each week. Since we have eight weeks, that means you should make 3 posts to the Activity Log each week, for a total of at least 24 posts. Each post will be worth 1 point.*

*If you are working as part of a team, make sure you clearly identify which team member worked on which tasks. The Activity Log should help me figure out how each team member contributed to the project. If I cannot discern who worked on what aspects of the project from the activity log, no points will be awarded for it.*

*Total points possible: 24*

Srinivasa Rao decided the data.. Data description was also completed

Srinivasa Rao found the data online

Pranay Reddy Lingapuram researched and described the techniques to store data in relational databases

Srinivasa Rao completed design process.

Pranay Reddy Lingapuram did DDL tasks

Srinivasa Rao completed half of DML commands

Pranay Reddy completed the other half of DML

Srinivasa Rao defined and demonstrated indexes.

Pranay Reddy used views on database.

Srinivasa Rao did triggers, transactions, future work

Prannay completed security, locking, and backup.

Srinivasa Rao used python to perform database operations.

Prannay and Srinivasa worked together on php website to perform search operation on it.