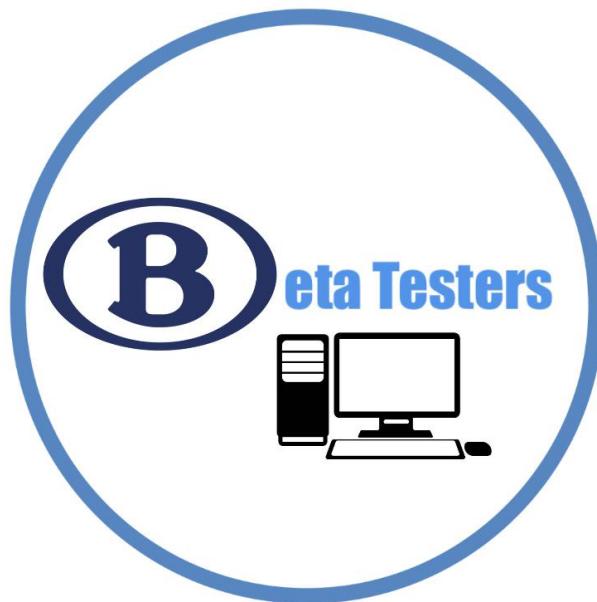


# Beta Testers



CIS360-28065-Spring22B

Team B

Christine Beck  
Jorge Casillas Castro  
Celeste Garcia  
Elijah Jackson  
Ryu Lewis  
Erika Moss  
Alec Valencia

# Executive Summary Team B - Beta Testers

## GOAL

- Our goal is to create a reliable database for Palermo theater, to accomplish this goal we will create a new design that will improve public relations and use our skills to format new data into the system

## COMPANY INFORMATION

- We as a company pride ourselves on creating new and effective databases that improve business flow and ensure to our clients that their data is in good hands. Our company has hands-on experience when redesigning new systems using the following procedures listed below.

## OUR PROCEDURES

- Beta Testers will create an ERD and Data Dictionary to show the client how their database will be formatted before it is implemented into the system
- Beta Testers will also use SQL to import the new data into the system
- We will use ACL testing to audit the new database for any errors in the system
- Access forms, reports, and Tableau visualizations will be used to show the data in an understandable format

## COMPANY SKILLS

- Beta Testers are a professional team that have experience when working and redesigning ineffective databases, skills include knowledge of SQL and organizational skill sets which every Beta Tester possess

## SUMMARY

- Beta Testers are capable of redesigning this database for Palermo theater and our proud to work for this company that will allow them to showcase their skills

# TABLE OF CONTENTS

<b>Executive Summary Team B - Beta Testers</b>	<b>1</b>
<b>TABLE OF CONTENTS</b>	<b>2</b>
<b>TEAM MEMBER INTRODUCTIONS</b>	<b>3</b>
<b>CLIENT INTRODUCTION</b>	<b>6</b>
<b>CURRENT SITUATION AND REQUIREMENTS</b>	<b>7</b>
<b>INITIAL ENTITY-RELATIONSHIP DIAGRAM AND DATA DICTIONARY</b>	<b>8</b>
<b>INTERMITTENT ER DIAGRAMS</b>	<b>14</b>
<b>FINAL ER DIAGRAM AND DATA DICTIONARY EXPLANATION</b>	<b>20</b>
<b>LOGICAL RELATIONAL SCHEMA</b>	<b>25</b>
<b>SCREEN CAPTURE OF TABLES IN SQL</b>	<b>26</b>
<b>SCREEN CAPTURE OF ACL TESTS AND RESULTS</b>	<b>37</b>
<b>TECHNICAL ISSUES, QUIRKS, CHALLENGES, LESSONS LEARNED</b>	<b>44</b>
<b>APPENDICES A: SQL</b>	<b>45</b>
<b>ORIGINAL TABLE AND INSERT VALUE SCRIPTS</b>	<b>45</b>
<b>TESTS AND QUERIES</b>	<b>82</b>
<b>(SCRIPTS &amp; SCREEN CAPTURE)</b>	<b>82</b>
How many customers watched the movie "Toy Story" in 2012? Include the customer first and last names. List the first name alphabetically.	93
What is the date of birth in the years 1990-1995 of customers who watched "Back to the future"?	95
A similar two INSERT statements executed in the opposite order should fail.	96
Query: Most popular inventory item	97
<b>TRIGGERS AND PROCEDURES</b>	<b>102</b>
<b>(SCRIPTS &amp; SCREEN CAPTURE)</b>	<b>102</b>
<b>TRANSACTIONS</b>	<b>115</b>
<b>(SCRIPTS &amp; SCREEN CAPTURE)</b>	<b>115</b>
<b>ACCESS: FORMS AND REPORTS</b>	<b>119</b>

<b>TABLEAU VISUALIZATION</b>	<b>141</b>
<b>APPENDICES B: TEAM MEMBER RESUMES AND TIME SHEETS</b>	<b>158</b>
<b>TEAM JOURNAL ENTRIES</b>	<b>166</b>
<b>APPENDICES C</b>	<b>188</b>
<b>TEAM CHARTER AND MEMBERSHIP AGREEMENT &amp;</b>	<b>188</b>
<b>TEAM MEETING AGENDAS</b>	<b>188</b>
<b>TEAM MEETING AGENDAS</b>	<b>191</b>

## TEAM MEMBER INTRODUCTIONS

We are the Beta Testers! Based in Arizona, this global team of up-and-coming self-starters are here to make your database management problems a thing of the past. No matter your goals, our team is here to guide, problem solve, and ultimately ease the burden of tracking any of your business's needs - your employees, customers, inventory - you name it, we can organize it for you! No task is too big nor too small for the Beta Testers. Whether you have one location or one-thousand locations, we understand that everyone's data is important, and having fast, easy access to your business's information need only be a click away.

Get to know our team, and know that you are in the right hands:

**CHRISTINE BECK:** After a decade of running a small, brick and mortar hobby store in Arizona, Christine Beck has returned to school to complete a Bachelor's in Accounting from Arizona State University. As a small-business owner she has learned to excel in many fields of business, and fully understands the importance of a well-working database. From sales and customer purchase tracking, to inventory and supply chain management, Christine enjoys the detail-oriented nature of data collection and organization.

**JORGE CASILLAS CASTRO:** Jorge Castro is a senior at ASU and is attending W.P. Carey School of Business in order to get a B.S in Accountancy. He is going to be pursuing a degree in law afterwards. In his free time he goes to the gym and works full time as well as takes care of his daughter. Previously he served five years in the Marine Corps where he was able to be

accountable for teams of four people to thirty at different times. He also planned training and was held accountable for maintaining records of his team. After the Marines he moved into Starbucks to become a shift supervisor where he was able to learn a deeper understanding of daily and weekly ordering to ensure high volume stores were always properly stock with inventory he was also granted the ability to interview and hire candidates, he was also responsible for training new shift supervisors to follow correct standards. Jorge has a good handle of working under pressure.

**CELESTE GARCIA:** After beginning her academic career at Arizona State University as an Accounting major, Celeste Garcia has worked diligently to expand her skills and knowledge of the finance and business industry. Her past work experience at Arizona State University Financial Aid and Scholarship Services office, and at the SEMTE Business office has fostered her interest in the growth and development of interpersonal and technical skills that are useful in the industry. Celeste plans to work closely with the client to provide strong attention to detail and clear communication between parties, to ensure the accomplishment of the project goals.

**ELIJAH JACKSON:** As a current Accounting major at Arizona State University, Elijah Jackson is eager to learn valuable skills and make professional connections that may benefit him in his future career path. Elijah plans on working with clients to ensure that they understand the basics of working with this database. Elijah looks forward to working with this team to accomplish the goals that have been set.

**RYU LEWIS:** As an accountancy major at Arizona State University, Ryu Lewis aspires to serve his client by developing and manipulating databases for them. During his teenage years, he has experience from working at a convenience store where he established ideal accounting by calculating daily sales and maintaining records. Ryu is looking forward to utilizing his skills to complete the project and create professional connections.

**ERIKA MOSS:** As a current accounting intern, Erika Moss is looking forward to serving the Palermo Theater Group and developing a professional database for them. Erika looks forward to helping the client achieve their goal of having an easy-to-use database. Erika knows the importance of collaboration and teamwork from her work as an intern. She has served other clients as an auditor and helped to review audit documentation to give clients an audit opinion.

**ALEC VALENCIA:** Alec Valencia is a senior at Arizona State University and is studying Information security at W.P. Carey School of Business. Alec is looking to enter the cybersecurity field after he graduates. Alec specializes in computer programming and enjoys coding and working on his own projects. Alec plans on completing a cybersecurity internship for the coffee company Starbucks in the summer and obtaining his security plus certification as well. Alec has

experience with Linux and penetration testing using Kali linux and Parrot OS. He also has experience with databases, SQL and Python.

## CLIENT INTRODUCTION

Founded in 2005, Palermo Theater Group is located here in Phoenix, Arizona and the surrounding area. Their business has blossomed quickly since their establishment. In the past 15 years, they have opened 12 theater locations around the valley. Each location has a specific theme based on a movie and is decorated to match characters and scenes from that movie. For example, one theater is patterned after *The Titanic* and another is inspired by *The Avengers*. These themes invite customers to take advantage of visiting each unique theater and getting to enjoy the decor at each one.



Palermo provides its customers with a luxurious movie-going experience. Each theater contains reclining seats, state-of-the-art screens, and a bar. There are a variety of drinks and concessions offered at the Snack Stand and Bar. Palermo offers the classic movie snacks such as popcorn and candy, but also offers a vegetable tray and a veggie burger. Each customer is valued for their individual needs and offered refreshments that match their personal lifestyle. Customers are able to enjoy their food and drinks while watching a movie.

Palermo rewards its customers with a loyalty program. The loyalty program has the following 5 reward levels: Emmy, Grammy, Oscar, Tony, and EGOT. Advancing to each reward level is based on the number of times a customer visits a theater within the year. Each level rewards the customer with one or two free items they can receive during a future visit. These rewards range from a free soda and popcorn up to free movie tickets and pizzas.

Palermo's goal is to make the movie-going experience luxurious and relaxed. They plan to serve the Phoenix area for many years to come!

## CURRENT SITUATION AND REQUIREMENTS

Palermo has developed a system called Aloha to handle the scheduling and accounting needs of each theater. The theater group has asked the Beta Testers to develop a database to track the loyalty program and the customers visiting the theaters. This database needs to cover the data from all 12 theater locations.

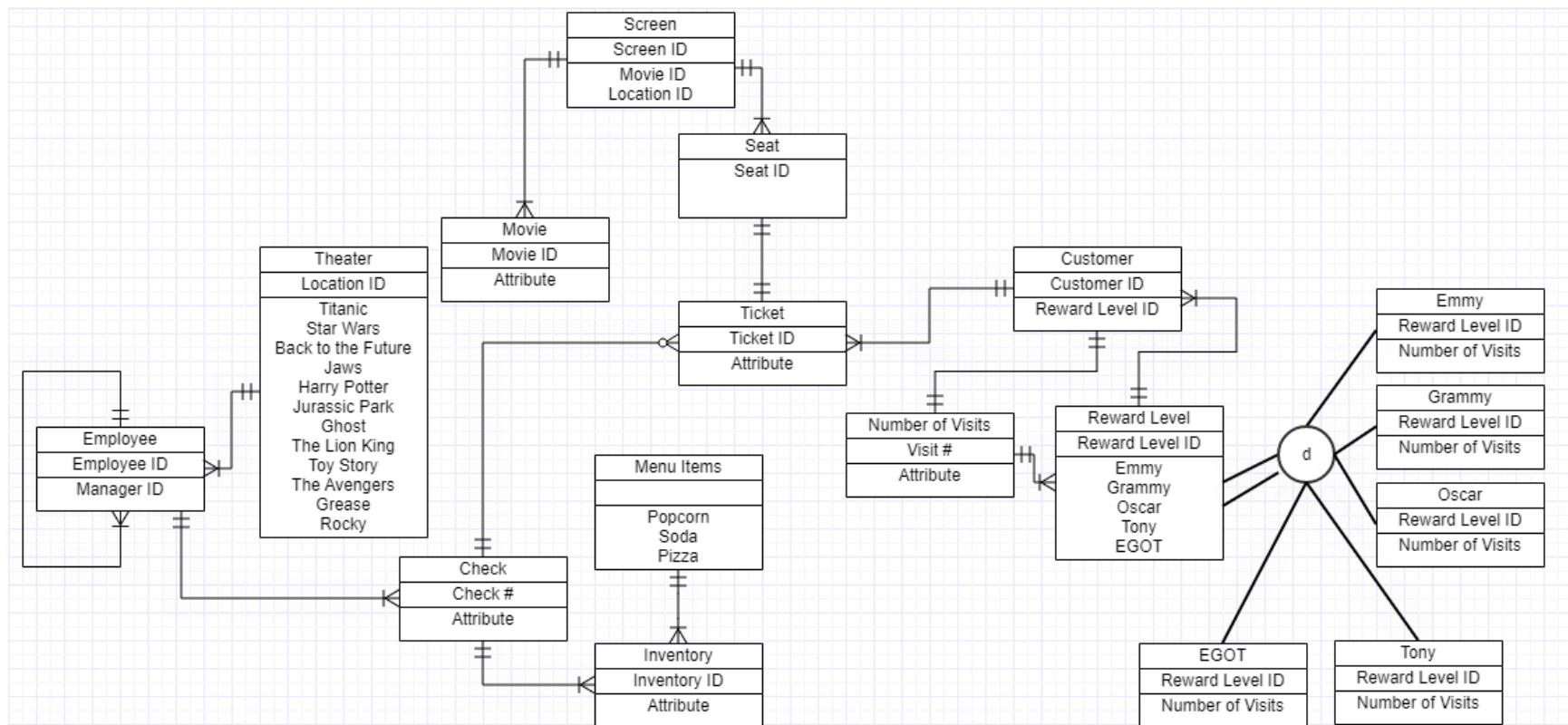
Palermo has also asked that the database improve their inventory system. Inventory is not being ordered in time, which is leading to a shortage of certain products at the theater. Inventory is being counted incorrectly. Having an automated system that would update inventory counts and notify Palermo when inventory needs to be ordered again is critical.

The database needs to track the customers and their loyalty level. The number of visits in a year needs to be recorded and the loyalty level needs to be updated at the appropriate visit number. The different loyalty levels have certain rewards associated with them. The database needs to trigger those rewards for the customer's next visit.

Palermo has asked for these components to be added to the database to increase the functionality and efficiency of the theaters.

# INITIAL ENTITY-RELATIONSHIP DIAGRAM AND DATA DICTIONARY

Figure 1.1  
Initial ERD with all entities and relationships



**Figure 1.2**  
Initial Data Dictionary

Table Name (Entity)	Attribute Name	Contents	Type	Format	Range	Required	PK or FK	FK Reference Table
Theatre	<b>LOCATION_ID</b>	Numeric Theatre Location ID	INTEGER	#####	1000-9999	Y	PK	
	NAME	Theatre Theme Name (ie. Titanic)	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	ADDRESS	Theatre Address	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	CITY	Theatre Address	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	STATE	Theatre Address	CHAR(2)	XX	A-Z	Y		
	ZIP	Theatre Address	CHAR(5)	#####	00000-99999	Y		
	PHN_NUM	Theatre Phone Number	CHAR(10)	### #### ####	N/A	Y		
Employee	<b>EMPLOYEE_ID</b>	Numeric ID Number	INTEGER	#####	1000-9999	Y	PK	
	F_NAME	Employee First Name	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	INITIAL	Employee Middle Initial	CHAR(1)	X	A-Z	Y		
	L_NAME	Employee Last Name	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	ADDRESS	Employee Address	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	CITY	Employee Address	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	STATE	Employee Address	CHAR(2)	XX	A-Z	Y		
	ZIP	Employee Address	CHAR(5)	#####	00000-99999	Y		
	PHN_NUM	Employee Phone Number	CHAR(10)	### #### ####	N/A	Y		
	EMAIL	Employee Email Address	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
Reward Level	<b>LOCATION_ID</b>	Numeric Theatre Location ID	INTEGER	#####	N/A	Y	FK	Theatre
	MANAGER_ID	Numeric Manager ID	INTEGER	#####	N/A	Y	FK	Employee
	<b>RWD_LVL_ID</b>	Reward Level #	INTEGER	#####	1000-9999	Y	PK	
	<b>NUM_VISIT</b>	Number of Annual Visits	MATHEMATICAL COMPUTATION	##	0-100	Y	FK	

Emmy	<b>RWD_LVL_ID</b>	Reward Level #	INTEGER	#####	1000-9999	Y	PKFK	Reward Level
	<i>NUM_VISIT</i>	Number of Annual Visits	MATHEMATICAL COMPUTATION	###	>=1	Y	FK	Visits
	EMMY	Name Of Reward	CHAR(4)	xxxx	A-Z	Y		
	<i>INVENTORY_ID</i>	Numeric Menu Item ID	INTEGER	#####	1000-9999	Y	FK	Inventory
Grammy	<b>RWD_LVL_ID</b>	Reward Level #	INTEGER	#####	1000-9999	Y	PKFK	Reward Level
	<i>NUM_VISIT</i>	Number of Annual Visits	MATHEMATICAL COMPUTATION	###	>=5	Y	FK	Visits
	GRAMMY	Name Of Reward	CHAR(6)	xxxxxx	A-Z	Y		
	<i>INVENTORY_ID</i>	Numeric Menu Item ID	INTEGER	#####	1000-9999	Y	FK	Inventory
Oscar	<b>RWD_LVL_ID</b>	Reward Level #	INTEGER	#####	10000-9999	Y	PKFK	Reward Level
	<i>NUM_VISIT</i>	Number of Annual Visits	MATHEMATICAL COMPUTATION	###	>=15	Y	FK	Visits
	OSCAR	Name Of Reward	CHAR(5)	xxxxx	A-Z	Y		
	<i>INVENTORY_ID</i>	Numeric Menu Item ID	INTEGER	#####	1000-9999	Y	FK	Inventory
Tony	<b>RWD_LVL_ID</b>	Reward Level #	INTEGER	#####	1000-9999	Y	PKFK	Reward Level
	<i>NUM_VISIT</i>	Number of Annual Visits	MATHEMATICAL COMPUTATION	###	>=30	Y	FK	Visits
	TONY	Name Of Reward	CHAR(4)	xxxx	A-Z	Y		
	<i>INVENTORY_ID</i>	Numeric Menu Item ID	INTEGER	#####	1000-9999	Y	FK	Inventory
EGOT	<b>RWD_LVL_ID</b>	Reward Level #	INTEGER	#####	1000-9999	Y	PKFK	Reward Level
	<i>NUM_VISIT</i>	Number of Annual Visits	MATHEMATICAL COMPUTATION	###	>=50	Y	FK	Vistis
	EGOT	Name Of Reward	VARCHAR(4)	xxxx	A-Z	Y		
	<i>INVENTORY_ID</i>	Numeric Menu Item ID	INTEGER	#####	1000-9999	Y	FK	Inventory
Visit	<b>NUM_VISIT</b>	Number of Annual Visits	MATHEMATICAL COMPUTATION	##	0-100	Y	PK	
Movie	<b>MOVIE_ID</b>	Numeric Movie ID	INTEGER	#####	1-9999999	Y	PK	

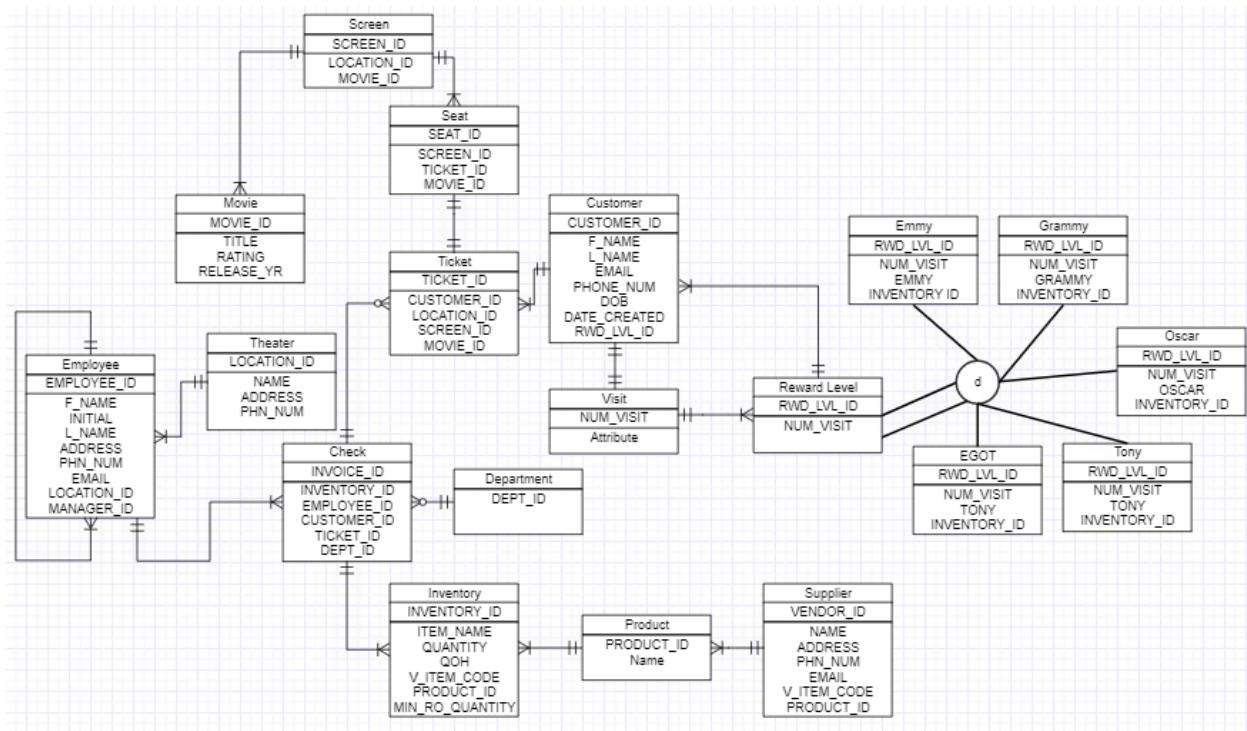
	TITLE	Movie Title	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	RATING	Movie Rating	VARCHAR(5)	Xxxxx	N/A	Y		
	RELEASE_YR	Movie Release Year	DATE	DD-MON-YYYY	N/A	Y		
Customer	<b>CUSTOMER_ID</b>	Numeric Customer ID	INTEGER	#####	1000-9999		PK	
	F_NAME	Customer First Name	VARCHAR(35)	Xxxxxxxxxxxxxxx	N/A	Y		
	L_NAME	Customer Last Name	VARCHAR(35)	Xxxxxxxxxxxxxxx	N/A	Y		
	EMAIL	Customer Email	VARCHAR(35)	Xxxxxxxxxxxxxxx	N/A	Y		
	PHONE_NUM	Customer Phone Number	CHAR(10)	### #### ####	N/A	Y		
	DOB	Customer Date of Birth	DATE	DD-MON-YYYY	N/A	Y		
	DATE_CREATED	Date Of Customer ID Creation	DATE	DD-MON-YYYY	N/A	Y		
	NUM_VISIT	Number of Annual Visits	INTEGER	##	0-100	Y	FK	Visit
	RWD_LVL_ID	Numeric Reward Level	INTEGER	#####	N/A	Y	FK	Reward Level
Ticket	<b>TICKET_ID</b>	Numeric Ticket Number	INTEGER	#####	1000-9999	Y	PK	
	CUSTOMER_ID	Numeric Customer ID	INTEGER	#####	1000-9999	N	FK	Customer
	LOCATION_ID	Numeric Location ID	INTEGER	#####	1000-9999	Y	FK	Theatre
	SCREEN_ID	Numeric Screen ID	INTEGER	#####	1000-9999	Y	FK	Screen
	MOVIE_ID	Numeric Movie ID	INTEGER	#####	1000-9999	Y	FK	Movie
	SEAT_ID	Alpha-Numeric Seat Number	VARCHAR(5)	Xxxxx	N/A	Y	FK	Seat
Screen	<b>SCREEN_ID</b>	Numeric Screen ID	INTEGER	#####	1000-9999	Y	PK	
	LOCATION_ID	Numeric Location ID	INTEGER	#####	1000-9999	Y	FK	Theatre
	MOVIE_ID	Numeric Movie ID	INTEGER	#####	1000-9999	Y	FK	Movie
Inventory	<b>INVENTORY_ID</b>	Numeric Menu Item	INTEGER	#####	1000-9999	Y	PK	
	ITEM_NAME	Item Name	VARCHAR(35)	#####	N/A	Y		
	QOH	Quantity on Hand	SMALLINT	#####	0-9999	Y		

	<b>MIN_RO_QUANT</b>	Minumum Reorder Quantity	SMALLINT	#####	0-9999	Y		
	<b>PRICE</b>	Menu Item Price to Customer	DECIMAL (3,2)	0.##	0-9	Y		
	<b>V_ITEM_CODE</b>	Vendor Item Number	VARCHAR(35)	Xxxxxxxxxx	N/A	Y	FK	Supplier
	<b>PRODUCT_ID</b>	Numeric Product ID Number	INTEGER	#####	1000-9999	Y	FK	Product
Product	<b>PRODUCT_ID</b>	Numeric Product ID Number	CHAR(10)	#####	0-9999999999	Y	PK	
	<b>PROD_NAME</b>	Product Name	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	<b>PROD_SIZE</b>	Product Size (if applicable)	VARCHAR(35)	Xxxxxxxxxx	N/A	N		
	<b>PROD_COST</b>	Product Cost from Supplier	DECIMAL (8,2)	0.##	0-9	Y		
	<b>V_ITEM_CODE</b>	Vendor Item Number	VARCHAR(35)	Xxxxxxxxxx	N/A	Y	FK	Supplier
Department	<b>DEPT_ID</b>	Numeric Department ID Number	INTEGER	#####	1000-9999	Y	PK	
Supplier	<b>VENDOR_ID</b>	Numeric Vendor ID Number	CHAR(6)	#####	1000-9999	Y	PK	
	<b>V_NAME</b>	Supplier Name	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	<b>V_ADDRESS</b>	Supplier Address	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	<b>V_CITY</b>	Supplier Address	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	<b>V_STATE</b>	Supplier Address	CHAR(2)	XX	A-Z	Y		
	<b>V_ZIP</b>	Supplier Address	CHAR(5)	####	00000-99999	Y		
	<b>V_PHN_NUM</b>	Supplier Phone Number	CHAR(10)	#### #### ####	N/A	Y		
	<b>V_EMAIL</b>	Supplier Email	VARCHAR(35)	Xxxxxxxxxx	N/A	Y		
	<b>V_CONTACT</b>	Supplier Point of Contact	VARCHAR(35)	Xxxxxxxxxx	N/A	N		
	<b>V_ITEM_CODE</b>	Vendor Item Number	INTEGER	#####	1000-999	Y		
	<b>PRODUCT_ID</b>	Numeric Product ID Number	INTEGER	#####	1000-9999	Y	FK	Product
Check	<b>MENU_ITEM_ID</b>	Numeric Invoice ID	INTEGER	#####	1000-9999	Y	PK	
	<b>INVENTORY_ID</b>	Numeric Menu Item	INTEGER	#####	1000-9999	N	FK	Inventory
	<b>QUANTITY</b>	Quantity Purchased by Customer	SMALLINT	#####	0-9999	N		

	<i>PRICE</i>	Menu Item Price to Customer	DECIMAL (3,2)	0##	0-9	N	FK	Inventory
	<i>EMPLOYEE_ID</i>	Numeric Employee ID	INTEGER	#####	1000-9999	Y	FK	Employee
	<i>CUSTOMER_ID</i>	Numeric Customer ID	INTEGER	#####	1000-9999	N	FK	Customer
	<i>TICKET_ID</i>	Numeric Ticket ID	INTEGER	#####	1000-9999	N	FK	Ticket
	<i>DEPT_ID</i>	Numeric Department ID	INTEGER	#####	1000-9999	Y	FK	Department
	<i>LOCATION_ID</i>	Numeric Theatre Location ID	INTEGER	#####	1000-9999	Y	FK	Theatre
<hr/>								
Seats	<b>SEAT_ID</b>	Alpha-Numeric Seat Number	VARCHAR(5)	Xxxxx	N/A	Y	PK	
	<i>SCREEN_ID</i>	Numeric Screen ID	INTEGER	#####	1000-9999	Y	FK	Screen
	<i>TICKET_ID</i>	Numeric Ticket Number	INTEGER	#####	1000-9999	Y	FK	Ticket
	<i>MOVIE_ID</i>	Numeric Movie ID	INTEGER	#####	1000-9999	Y	FK	Movie

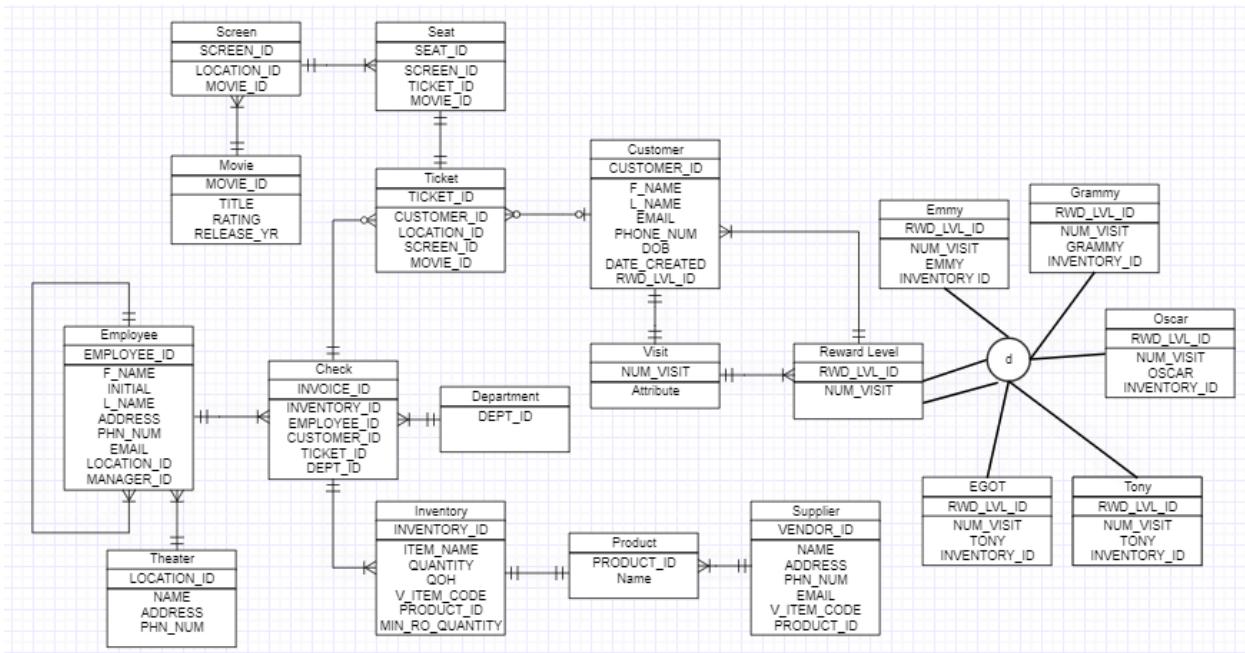
# INTERMITTENT ER DIAGRAMS

Figure 2.1



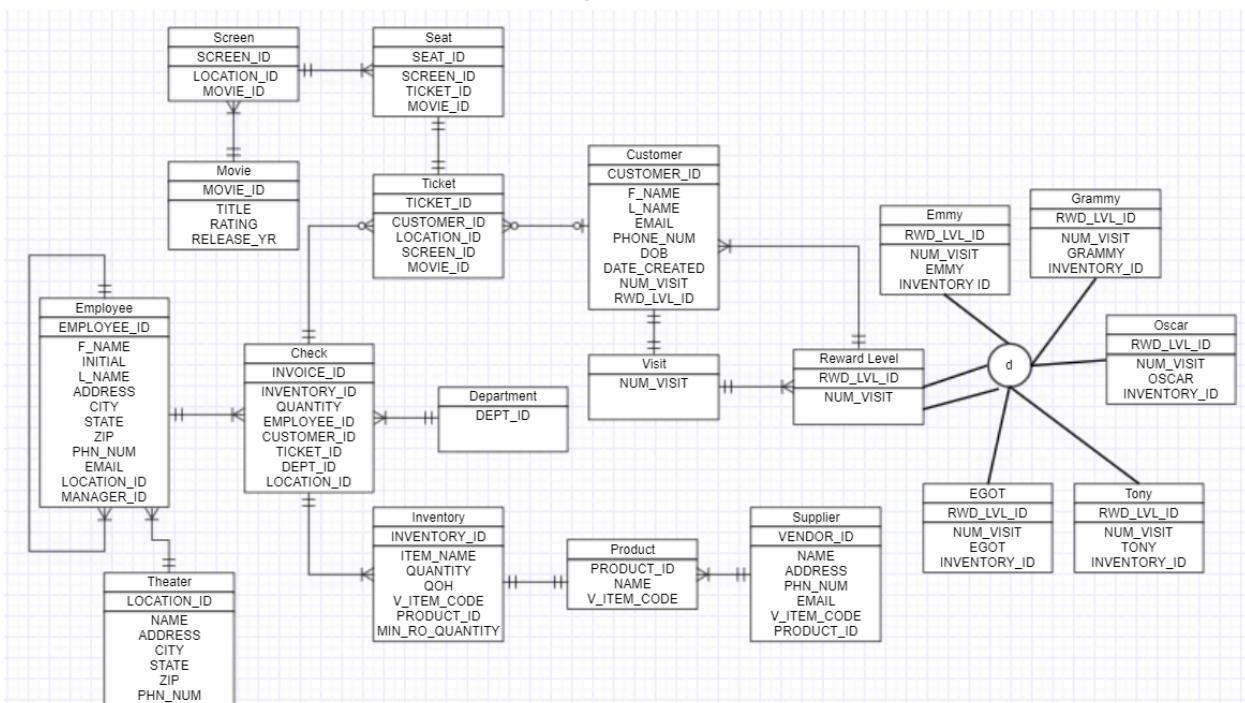
Updated attributes on Employee, Theatre, etc.

Figure 2.2



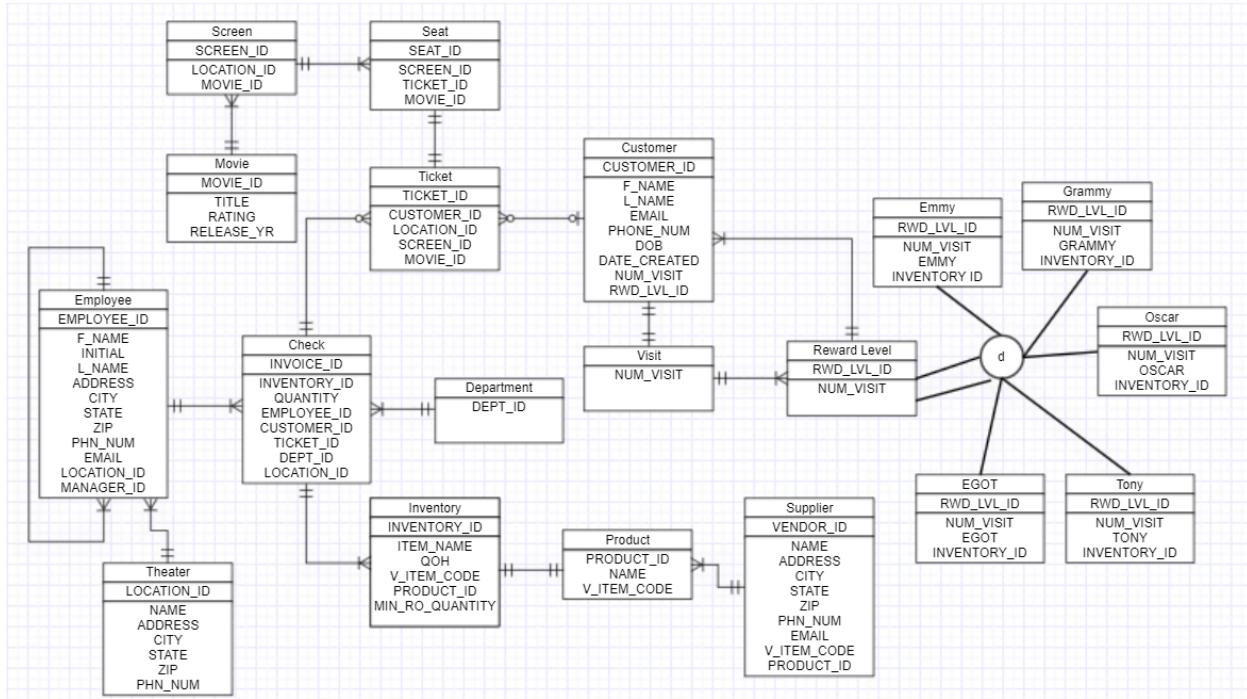
Added table for Screen

Figure 2.3



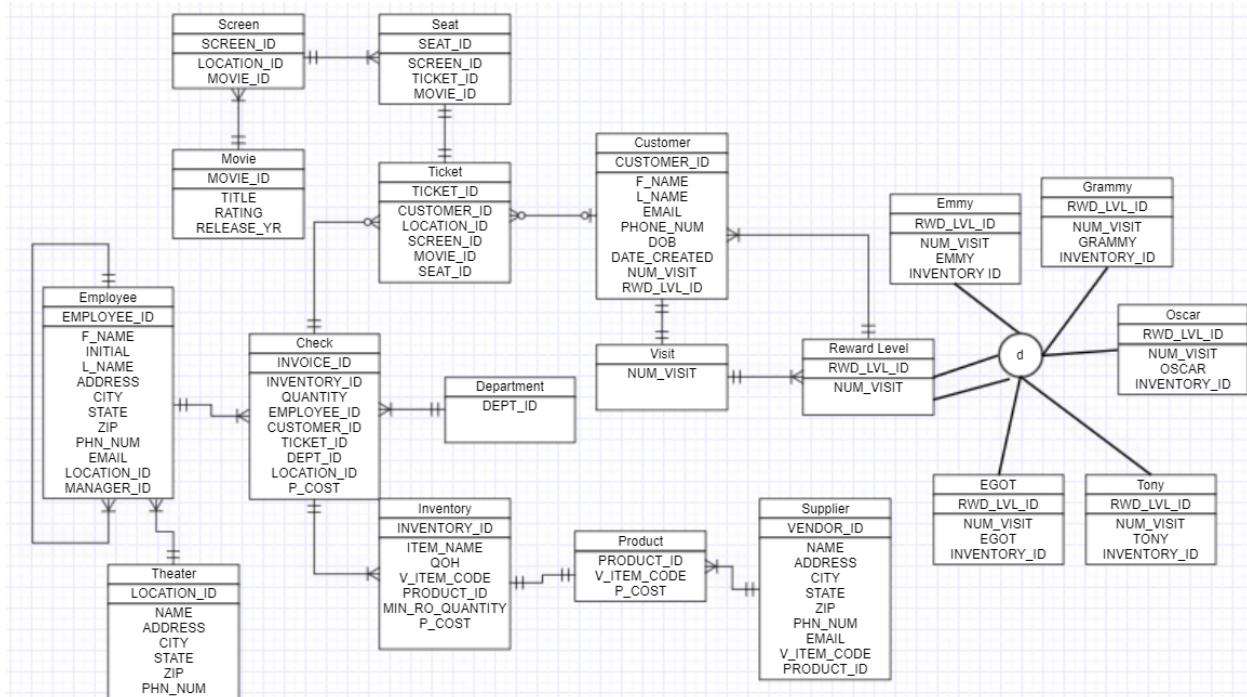
Updated Theatre, Product, and Customer tables

Figure 2.4



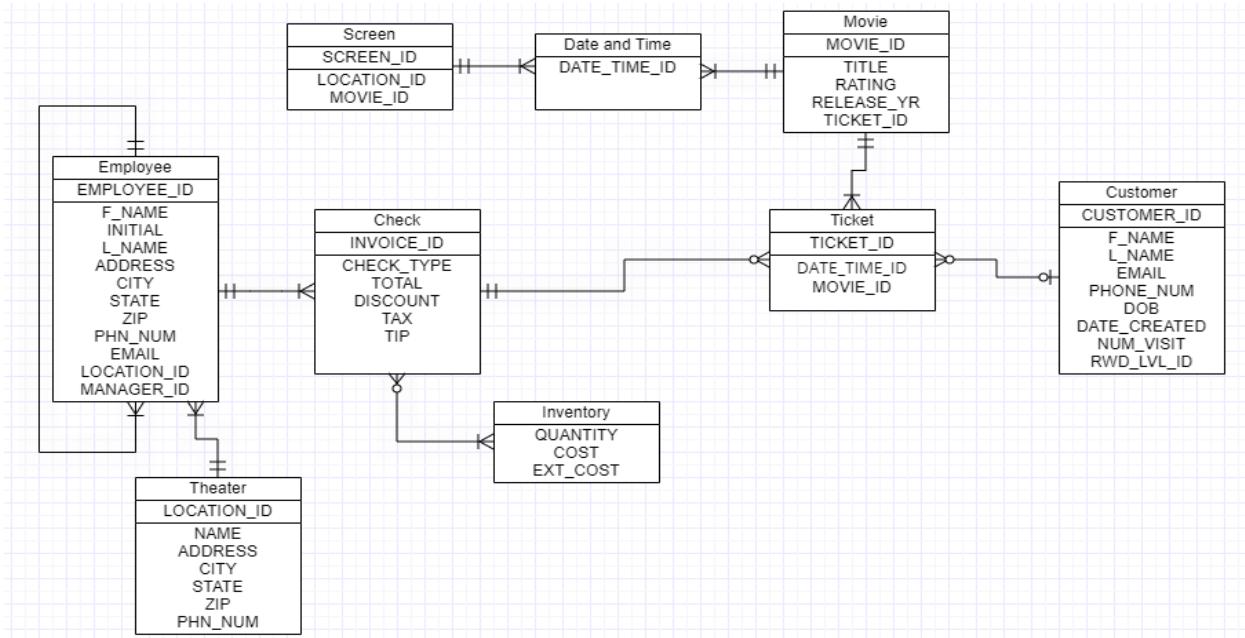
Updated Supplier and Inventory tables

Figure 2.5



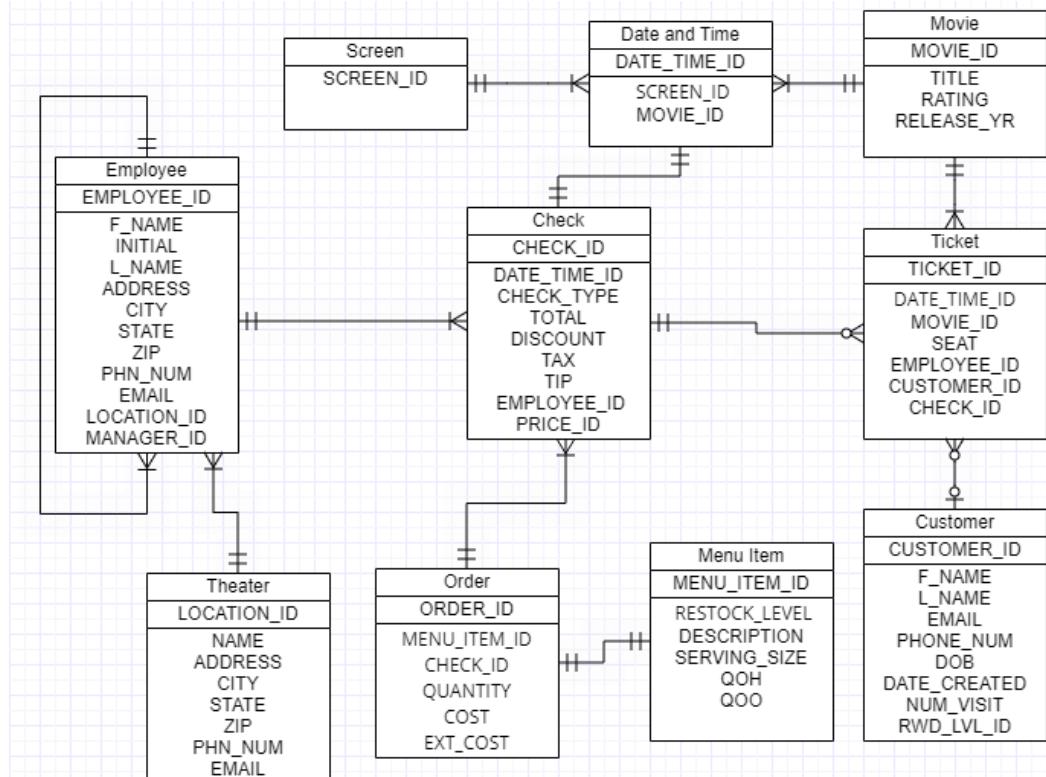
Updating Inventory, Check, and Ticket

Figure 2.6



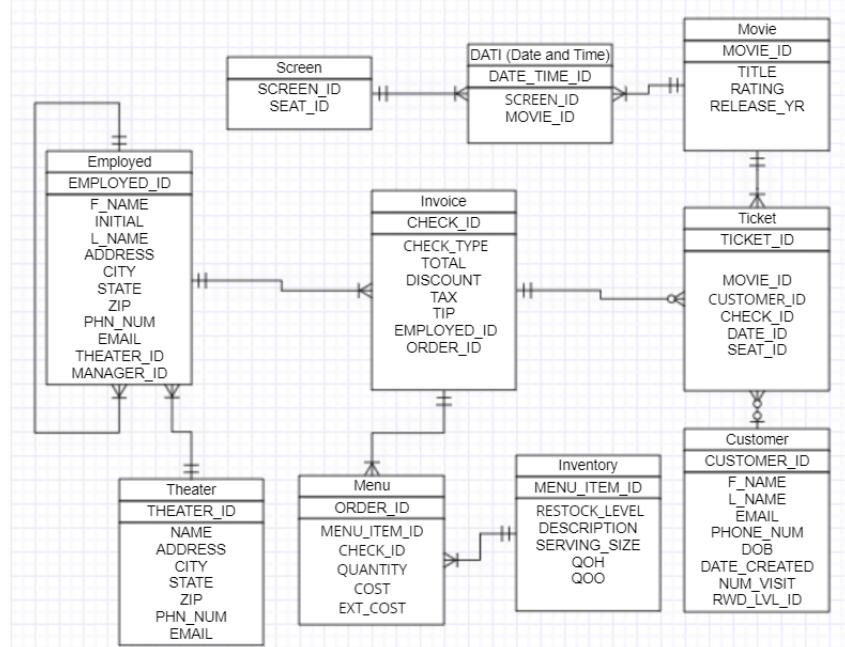
Making updates based on Milestone 3 feedback

Figure 2.7



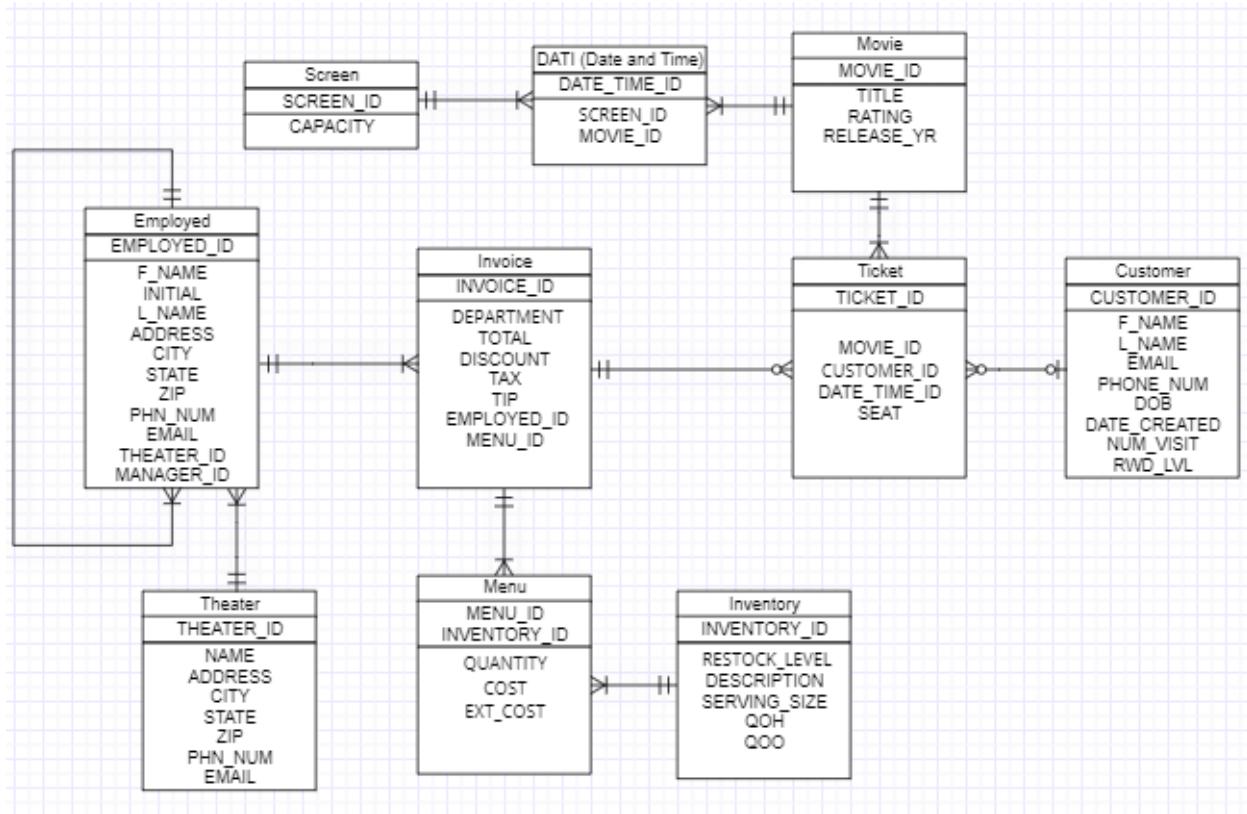
Updating all tables (except Employee) based on feedback

Figure 2.8



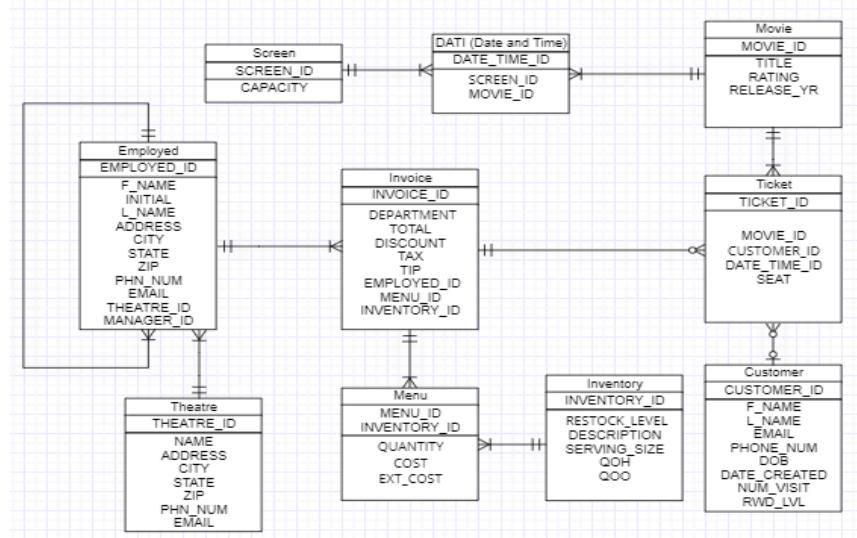
Update and rename tables

Figure 2.9



Updating table attributes

Figure 2.10



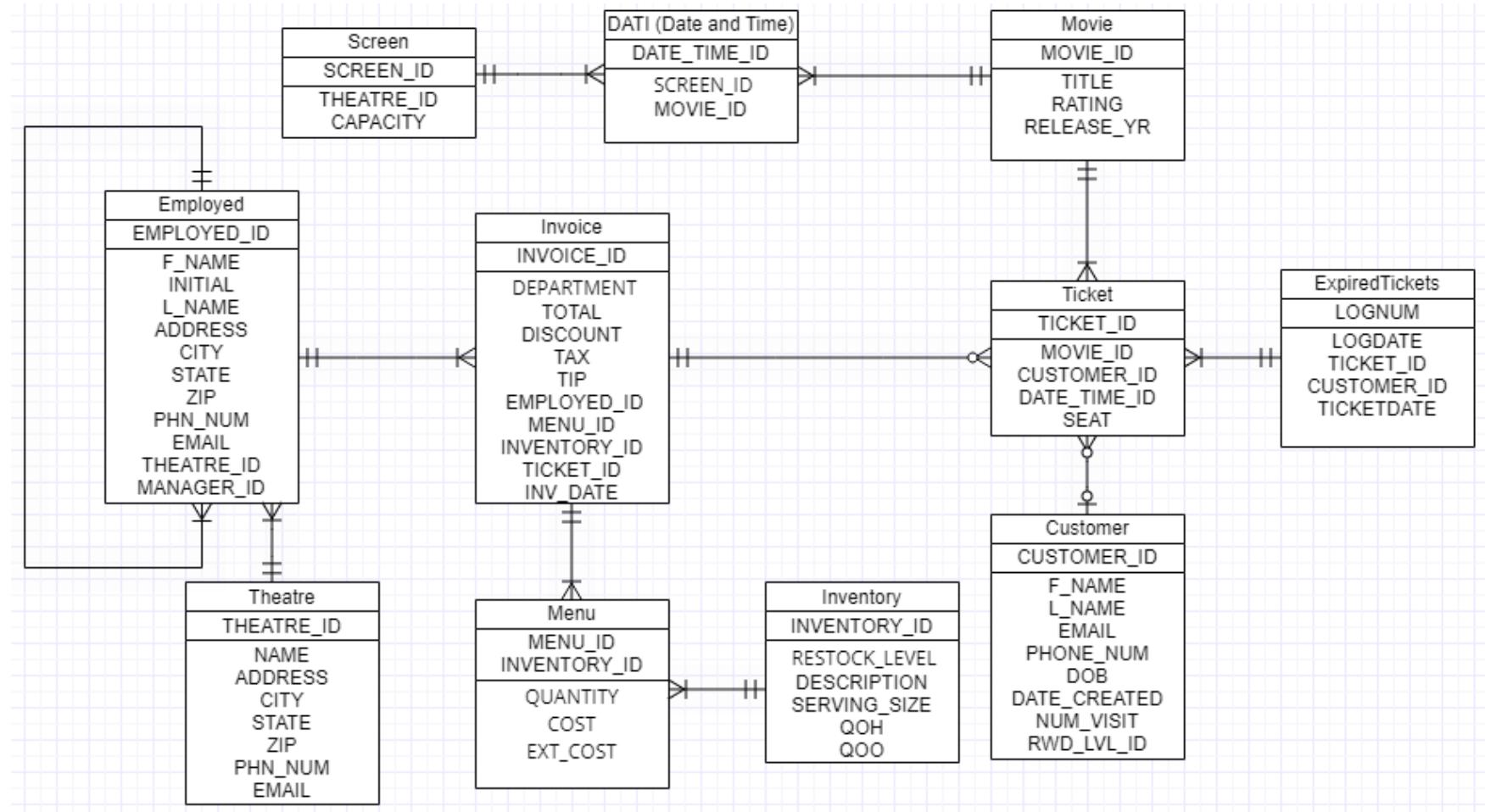
Updating table names and attributes to be consistent with the database

## FINAL ER DIAGRAM AND DATA DICTIONARY EXPLANATION

Final ERD (pictured below) showing all entities input into the Database. This revised ERD contains all attributes and entities that are in the SQL database.

Employees of Palermo are assigned to a theatre to work at. The invoices are tied to the employees by their employee ID. The invoices contain inventory purchased and the total, discount, tax and tip. The menu contains the quantity purchased as well as the total cost for that transaction. Inventory contains the names of all the options at the Snack Stand and Bar. The movie tickets are related to the movie, customer, and expired tickets. The tickets purchased by the customer are used to track their number of visits to determine their reward level. The tickets are used to show how many tickets were purchased for a movie. At the end of the year, the tickets “expire” so that the number of visits can be recounted each year. The movies are identified by a MOVIE\_ID. This is tied to a showing time, or the Date and Time table, to reflect when the movie was shown. The showings are tied to a screen in a specific theatre.

Figure 3.1



Final ERD

**Figure 3.2**  
**Final Data Dictionary**

Screen	<b>SCREEN_ID</b>	Numeric Screen ID	CHAR(6)	#####	1000-9999	Y	PK	
	CAPACITY	Screen Capacity	CHAR(3)	###	0-250	Y		
	THEATRE_ID	Numeric Theatre ID	CHAR(3)	###	100-112	Y	FK	Theatre
Date and Time	<b>DATE_TIME_ID</b>	Numeric Screen ID	CHAR(14)	##############	N/A	Y	PK	
	SCREEN_ID	Numeric Screen ID	CHAR(3)	Xxx	100-200	Y		
	MOVIE_ID	Alpha-Numeric Movie ID	VARCHAR(6)	Xxxxxx	N/A	Y		
Movie	<b>MOVIE_ID</b>	Numeric Movie ID	VARCHAR(6)	#####	1-9999999	Y	PK	
	TITLE	Movie Title	VARCHAR(35)	Xxxxxxxxxxxx	N/A	Y		
	RATING	Movie Rating	VARCHAR(5)	Xxxxx	N/A	Y		
	RELEASE_YR	Movie Release Year	DATE	DD-MON-YYYY	N/A	Y		
Ticket	<b>TICKET_ID</b>	Numeric Ticket ID	CHAR(6)	#####	100000-999999	Y	PK	
	MOVIE_ID	Numeric Movie ID	VARCHAR(6)	#####	A10000-A99999	Y	FK	Movie
	CUSTOMER_ID	Numeric Customer ID	CHAR(6)	#####	000001-999999		PK	
	DATE_TIME_ID	Numeric Date Time ID	CHAR(14)	##############	N/A	Y	FK	Date and Time
	SEAT	Alpha-Numeric Seat Number	VARCHAR(3)	#####	900000-999999	Y	FK	
Customer	<b>CUSTOMER_ID</b>	Numeric Customer ID	CHAR(6)	#####	100000-999999		PK	
	F_NAME	Customer First Name	VARCHAR(35)	Xxxxxxxxxxxxxx	N/A	Y		
	L_NAME	Customer Last Name	VARCHAR(35)	Xxxxxxxxxxxxxx	N/A	Y		
	EMAIL	Customer Email	VARCHAR(35)	Xxxxxxxxxxxxxx	N/A	Y		
	PHONE_NUM	Customer Phone Number	CHAR(10)	#### #### ####	N/A	Y		

	DOB	Customer Date of Birth	DATE	YYYY-MM-DD	N/A	Y		
	DATE_CREATED	Date Of Customer ID Creation	DATE	YYYY-MM-DD	N/A	Y		
	NUM_VISIT	Number of Annual Visits	CHAR(3)	###	0-100	Y	FK	
	RWD_LVL_ID	Numeric Reward Level	CHAR(6)	#####	N/A	Y	FK	Reward Level
Invoice	<b>INVOICE_ID</b>	Numeric Menu ID	CHAR(6)	#####	200000-999999		PK	
	DEPARTMENT	Department Name	VARCHAR(35)	Xxxxxxxxxxx	N/A	Y		
	TOTAL	Total	DECIMAL (3,2)	0.##	0-9	Y		
	DISCOUNT	Discount	PERCENT	0.##	0-100	N		
	TAX	Tax	DECIMAL (3,2)	0.##	0-9	Y		
	TIP	Tip	DECIMAL (3,2)	0.##	0-9	N		
	EMPLOYED_ID	Numeric Employee ID	CHAR(6)	#####	000001-999999	Y	FK	Employee
	MENU_ID	Numeric Menu ID	CHAR(6)	#####	600000-999999	Y	FK	Menu
	INVENTORY_ID	Numeric Inventory ID	CHAR(3)	##	1-999	Y	FK	Inventory
	TICKET_ID	Numeric Ticket ID	CHAR(6)	#####	100000-999999	Y	PK	
	INV_DATE	Invoice Date	VARCHAR(10)	#####	N/A	Y		
Menu	<b>MENU_ID</b>	Numeric Menu ID	CHAR(6)	#####	600000-999999	Y	PK	
	INVENTORY_ID	Numeric Inventory ID	CHAR(3)	##	1-999	Y	PK	
	QUANTITY	Menu Item Quantity	CHAR(4)	#####	0-9999	N		
	COST	Menu Item Cost	DECIMAL (3,2)	##.##	0-9	Y		
	EXT_COST	Menu Item Extended Cost	DECIMAL (3,2)	##.##	0-9	Y		
Employee	<b>EMPLOYED_ID</b>	Numeric Employee ID	CHAR(6)	#####	000001-999999	Y	PK	
	F_NAME	Employee First Name	VARCHAR(35)	Xxxxxxxxxxx	N/A	Y		
	INITIAL	Employee Middle Initial	CHAR(1)	X	A-Z	Y		
	L_NAME	Employee Last Name	VARCHAR(35)	Xxxxxxxxxxx	N/A	Y		

	ADDRESS	Employee Address	VARCHAR(35)	Xxxxxxxxxxx	N/A	Y		
	CITY	Employee Address	VARCHAR(35)	Xxxxxxxxxxx	N/A	Y		
	STATE	Employee Address	CHAR(2)	XX	A-Z	Y		
	ZIP	Employee Address	CHAR(5)	#####	0-99999	Y		
	PHN_NUM	Employee Phone Number	CHAR(10)	### #### ####	N/A	Y		
	EMAIL	Employee Email Address	VARCHAR(35)	Xxxxxxxxxxxx	N/A	Y		
	THEATRE_ID	Numeric Theatre ID	CHAR(3)	###	100-112	Y	FK	Theatre
	MANAGER_ID	Numeric Manager ID	CHAR(6)	#####	N/A	Y	FK	Employee
Theatre	THEATRE_ID	Numeric Theatre ID	CHAR(6)	###	100-112	Y	PK	
	NAME	Theatre Theme Name (ie. Titanic)	VARCHAR(35)	Xxxxxxxxxxx	N/A	Y		
	ADDRESS	Theatre Address	VARCHAR(35)	Xxxxxxxxxxx	N/A	Y		
	CITY	Theatre Address	CHAR(35)	Xxxxxxxxxxx	A-Z	Y		
	STATE	Theatre Address	CHAR(2)	XX	A-Z	Y		
	ZIP	Theatre Address	CHAR(5)	#####	0-99999	Y		
	PHN_NUM	Theatre Phone Number	CHAR(10)	### #### ####	N/A	Y		
	EMAIL	Theatre Email	VARCHAR(35)	Xxxxxxxxxxxxxx	N/A	Y		
Inventory	INVENTORY_ID	Numeric Inventory ID	CHAR(3)	###	001-999	Y	PK	
	RESTOCK_LEVEL	Restock Level	CHAR(5)	#####	00001-99999	Y		
	DESCRIPTION	Item Description	VARCHAR(35)	Xxxxxxxxxxx	N/A	Y		
	SERVING_SIZE	Item Serving Size	VARCHAR(35)	Xxxxxxxxxxx	N/A	N		
	QOH	Quantity on Hand	CHAR(4)	####	0-9999	Y		
	QOO	Quantity on Order	CHAR(4)	####	0-10000	Y		

## LOGICAL RELATIONAL SCHEMA

**EMPLOYEE**(PK EMPLOYED\_ID, F\_NAME, INITIAL, L\_NAME, ADDRESS, CITY, STATE, ZIP, PHN\_NUM, EMAIL, FK THEATER\_ID, FK MANAGER\_ID)

**THEATRE**(PK THEATRE\_ID, NAME, ADDRESS, CITY, STATE, ZIP, PHN\_NUM, EMAIL)

**INVOICE**(PK INVOICE\_ID, DEPARTMENT, TOTAL, DISCOUNT, TAX, TIP, FK EMPLOYED\_ID, FK MENU\_ID, FK INVENTORY\_ID, FK TICKET\_ID, INV\_DATE)

**MENU**(PK MENU\_ID, PK INVENTORY\_ID, QUANTITY, COST, EXT\_COST)

**INVENTORY**(PK INVENTORY\_ID, RESTOCK\_LEVEL, DESCRIPTION, SERVING\_SIZE, QOH, QOO)

**TICKET**(PK TICKET\_ID, FK MOVIE\_ID, FK CUSTOMER\_ID, FK DATE\_TIME\_ID, SEAT)

**CUSTOMER**(PK CUSTOMER\_ID, F\_NAME, L\_NAME, EMAIL, PHONE\_NUM, DOB, DATE\_CREATED, NUM\_VISIT, RWD\_LVL\_ID)

**MOVIE**(PK MOVIE\_ID, TITLE, RATING, RELEASE\_YR)

**DATI(DATE AND TIME)**(PK DATE\_TIME\_ID, FK SCREEN\_ID, FK MOVIE\_ID)

**SCREEN**(PK SCREEN\_ID, FK THEATRE\_ID, CAPACITY)

**EXPIRED TICKETS**(PK LOGNUM, FK LOGDATE, FK TICKET\_ID, FK CUSTOMER\_ID, TICKETDATE)

# SCREEN CAPTURE OF TABLES IN SQL

SQLQuery1.sql - wp..UAD\jecasil1 (106)\* → X

```
Select *
From Theatre
```

100 %

Results Messages

	THEATRE_ID	NAME	ADDRESS	CITY	STATE	ZIP	PHN_NUM	EMAIL
1	101	TITANIC	1912 S. APRIL AVE.	PHOENIX	AZ	85001	6025557865	TITANIC@PALERMO.COM
2	102	STAR WARS	52577 N. LUCAS LN.	SCOTTSDALE	AZ	85054	4805555334	STARWARS@PALERMO.COM
3	103	BACK TO THE FUTURE	85515 W. MCFLY RD.	TEMPE	AZ	85202	6025552225	BACK2FUTURE@PALERMO.COM
4	104	JAWS	62075 S. SPIELBERG ST.	FOUNTAIN HILLS	AZ	85268	4805556722	JAWS@PALERMO.COM
5	105	HARRY POTTER	111401 W. PRIVET DR.	CHANDLER	AZ	85224	4805554774	HARRYPOTTER@PALERMO.COM
6	106	JURASSIC PARK	15 W. PREHISTORIC PKWY.	MESA	AZ	85201	4805553466	DINOPARK@PALERMO.COM
7	107	GHOST	71390 N. SPECTER ST.	GLENDALE	AZ	85301	6235553232	GHOST@PALERMO.COM
8	108	THE LION KING	1994 E. PRIDE ROCK BLVD.	PEORIA	AZ	85301	6025557865	LIONKING@PALERMO.COM
9	109	TOY STORY	95113-A S. PIXAR PL.	SURPRISE	AZ	85335	6235552639	TOYSTORY@PALERMO.COM
10	110	THE AVENGERS	20126 E. ASSEMBLE AVE.	QUEEN CREEK	AZ	85140	4805557826	AVENGERS@PALERMO.COM
11	111	GREASE	7858 E. LIGHTENING LN.	GOODYEAR	AZ	85326	6235555683	GREASE@PALERMO.COM
12	112	ROCKY	19768 N. ADRIAN AVE.	GILBERT	AZ	85142	4805557865	ROCKY@PALERMO.COM

Query executed successfully. wpcmssql.cloud.wpcarey.asu... ASUAD\jecasil1 (106) BonlineB360 00:00:00 12 rows

Theatre Table (Figure 4.1)

SQLQuery2.sql - wp..UAD\jecasil1 (111)\* → X SQLQuery1.sql - wp..UAD\jecasil1 (106)\*

```
Select *
From EMPLOYED
```

100 %

Results Messages

	EMPLOYED_ID	F_NAME	INITIAL	L_NAME	ADDRESS	CITY	STATE	ZIP	PHN_NUM	EMAIL	MANAGER_ID	theatre_id
1	000001	SARA	R	JONES	1234 W MAIN ST.	TEMPE	AZ	85001	6023457899	SARAH.JONES@GMAIL.COM	NULL	112
2	000002	JANE	P	LEE	1231 S GEORGIA ST.	TEMPE	AZ	85001	4807678544	JANE.LEE@MAIL.COM	NULL	111
3	000003	ANNE	E	PRIOR	2405 E WARNER ST.	TEMPE	AZ	85003	4807678574	ANNE.PRIOR@GMAIL.COM	NULL	110
4	000004	THOMAS	M	SMITH	2468 S LOVEBIRD ST.	TEMPE	AZ	85005	4808956677	THOMAS.SMITH@GMAIL.COM	NULL	109
5	000005	JACOB	S	MILLER	2440 E 6ATH ST.	TEMPE	AZ	85005	6027676544	JACOB.MILLER@GMAIL.COM	NULL	108
6	000006	ELI	D	JOHNSON	1807 W PEPPER ST.	TEMPE	AZ	85005	4808753395	ELI.JOHNSON@GMAIL.COM	NULL	107
7	000007	SANTIAGO	B	REES	1446 E GUADALUPE ST.	TEMPE	AZ	85003	6024102347	SANTIAGO.REES@GMAIL.COM	NULL	106
8	000008	LLILY	H	COOK	1447 E TERESA ST.	TEMPE	AZ	85004	6028534599	LLILY.COOK@GMAIL.COM	NULL	105
9	000009	ANDREW	W	JAMES	1506 S TEMPE ST.	TEMPE	AZ	85004	6025679034	ANDREW.JAMES@GMAIL.COM	NULL	104
10	000010	PHOEBE	R	MOSS	2891 E CARMEL ST.	TEMPE	AZ	85005	4807678564	PHOEBE.MOSS@GMAIL.COM	NULL	103
11	000011	PIPER	L	LUND	1698 W GOLD ST.	TEMPE	AZ	85001	6026782091	PIPER.LUND@GMAIL.COM	NULL	102
12	000012	PENELOPE	O	IRELAND	1698 W GOLD ST.	TEMPE	AZ	85003	6029007509	PENELOPE.IRELAND@GMAIL.COM	NULL	101
13	000013	JOHN	A	NEWT	1534 E 34TH ST.	TEMPE	AZ	85001	6029980539	JOHN.NEWT@GMAIL.COM	000012	101
14	000014	ELLE	J	MOOSE	1700 W STAR ST.	TEMPE	AZ	85002	6020035411	ELLE.MOOSE@GMAIL.COM	000012	101
15	000015	MICHAEL	L	SCARN	2788 E POOL ST.	TEMPE	AZ	85005	4807564429	MICHAEL.SCARN@GMAIL.COM	000011	102
16	000016	SKYE	O	CARLSON	1400 E O'PAL ST.	TEMPE	AZ	85002	6020035411	SKYE.CARLSON@GMAIL.COM	000010	103
17	000017	ARCHER	E	MONSON	2806 E LINDRICK ST.	TEMPE	AZ	85004	6029085564	ARCHER.MONSON@GMAIL.COM	000009	104
18	000018	LANE	B	DESPAIN	1399 W JUSTICE ST.	TEMPE	AZ	85004	6027859333	LANE.DESPAIN@GMAIL.COM	000008	105
19	000019	UMA	T	STRONG	1569 W STRAW ST.	TEMPE	AZ	85004	6029086112	UMA.STRONG@GMAIL.COM	000007	106
20	000020	BILLY	W	MCDON...	1776 E INDEPENDENC...	TEMPE	AZ	85005	6027089376	BILLY.MCDONALD@GMAIL.COM	000006	107
21	000021	MANNY	P	BLUE	2011 E YOUNG ST.	TEMPE	AZ	85002	6028906134	MANNY.BLUE@GMAIL.COM	000005	108
22	000022	ERIN	L	DESTINY	1592 S TRUE ST.	TEMPE	AZ	85002	6023851290	ERIN.DESTINY@GMAIL.COM	000004	109
23	000023	POLLY	B	MCARTHUR...	1224 E STRAWBERRY...	TEMPE	AZ	85001	4807162253	POLLY.MCARTHUR@GMAIL.COM	000003	110
24	000024	BRANDON	J	BUNCH	3907 W TREE ST.	TEMPE	AZ	85005	6026904321	BRANDON.BUNCH@GMAIL.COM	000002	111

Query executed successfully. wpcmssql.cloud.wpcarey.asu... ASUAD\jecasil1 (111) BonlineB360 00:00:00 41 rows

Employed Table (Figure 4.2)

SQLQuery2.sql - wp...UAD\jecasil1 (111)\* | SQLQuery1.sql - wp...UAD\jecasil1 (106)\*

Select \*  
From EMPLOYED

Results Messages

	EMPLOYED_ID	F_NAME	INITIAL	L_NAME	ADDRESS	CITY	STATE	ZIP	PHN_NUM	EMAIL	MANAGER_ID	theatre_id
19	000019	UMA	T	STRONG	1569 W STRAW ST.	TEMPE	AZ	85004	6029086112	UMA.STRONG@GMAIL.COM	000007	106
20	000020	BILLY	W	MCDON...	1776 E INDEPENDENC...	TEMPE	AZ	85005	6027089370	BILLY.MCDONALD@GMAIL.COM	000006	107
21	000021	MANNY	P	BLUE	2011 E YOUNG ST.	TEMPE	AZ	85002	6028905134	MANNY.BLUE@GMAIL.COM	000005	108
22	000022	ERIN	L	DESTINY	1592 S TRUE ST.	TEMPE	AZ	85002	6023851290	ERIN.DESTINY@GMAIL.COM	000004	109
23	000023	POLLY	B	MCART...	1224 E STRAWBERRY...	TEMPE	AZ	85001	4807162253	POLLY.MCARTHUR@GMAIL.COM	000003	110
24	000024	BRANDON	J	BUNCH	3907 W TREE ST.	TEMPE	AZ	85005	6026904321	BRANDON.BUNCH@GMAIL.COM	000002	111
25	000025	KATYA	S	BERRY	3400 S APPLETREE ST.	TEMPE	AZ	85004	6023011542	KATYA.BERRY@GMAIL.COM	000001	112
26	000026	STEVEN	C	AUSTIN	2300 N HOUSTON ST.	TEMPE	AZ	85004	4806709327	STEVEN.AUSTIN@GMAIL.COM	000002	111
27	000027	KATIE	R	CHRISTI...	2809 E COOKIE ST.	TEMPE	AZ	85002	6020997184	KATIE.CHRISTIANSEN@GMAIL.COM	000003	110
28	000028	NINA	W	BALLARD	4003 S LIME ST.	TEMPE	AZ	85001	6020021185	NINA.BALLARD@GMAIL.COM	000004	109
29	000029	GREG	F	RIGGS	1875 E RURAL ST.	TEMPE	AZ	85006	6028471904	GREG.RIGGS@GMAIL.COM	000005	108
30	000030	CRISTY	A	VICTOR	4560 E SILVER ST.	TEMPE	AZ	85002	4807264002	CRISTY.VICTOR@GMAIL.COM	000006	107
31	000031	STELLA	O	RICKS	3608 W YUMA ST.	TEMPE	AZ	85002	4800965383	STELLA.RICKS@GMAIL.COM	000007	106
32	000032	CARRIE	P	LION	3900 E BROADWAY ST.	TEMPE	AZ	85005	4809087796	CARRIE.LION@GMAIL.COM	000008	105
33	000033	BLAKE	R	GARRETT	900 E SYDNEY ST.	TEMPE	AZ	85004	480948372	BLAKE.GARRETT@GMAIL.COM	000009	104
34	000034	ANNIE	E	OWENS	4700 W MOON ST.	TEMPE	AZ	85002	6023459086	ANNIE.OWENS@GMAIL.COM	000010	103
35	000035	SUNNY	K	SPENCER	2740 E PILOT ST.	TEMPE	AZ	85005	4809382494	SUNNY.SPENCER@GMAIL.COM	000011	102
36	000036	FRED	G	BUBBLES	2668 W TEA ST.	TEMPE	AZ	85003	4808045829	FRED.BUBBLES@GMAIL.COM	000012	101
37	000037	HALEY	E	SANDERS	4508 S ROSE ST.	TEMPE	AZ	85005	4808903348	HALEY.SANDERS@GMAIL.COM	000012	101
38	000038	HARRY	R	GRAHAM	100 S IGLOO ST.	TEMPE	AZ	85005	4808007182	HARRY.GRAHAM@GMAIL.COM	000012	101
39	000039	CHARLIE	C	WALKER	7200 S BULLHORN ST.	TEMPE	AZ	85005	4801110972	CHARLIE.WALKER@GMAIL.COM	000011	102
40	000040	MIA	B	FOX	3705 E CALICO ST.	TEMPE	AZ	85005	4808084902	MIA.FOX@GMAIL.COM	000010	103
41	000041	JACQUEL...	M	QWERTY	4356 N WILLIAMS RD.	GILB...	AZ	85142	4809874353	JQWERTY@EMAIL.COM	000011	109

Query executed successfully. | wpcmssql.cloud.wpcarey.asu... | ASUAD\jecasil1 (111) | BonlineB360 | 00:00:00 | 41 rows

Employed Table (continued)

SQLQuery3.sql - wp...UAD\jecasil1 (118)\* | SQLQuery2.sql - wp...UAD\jecasil1 (111)\* | SQLQuery1.sql - wp...UAD\jecasil1 (106)\*

Select \*  
From CUSTOMER

Results Messages

	CUSTOMER_ID	F_NAME	L_NAME	EMAIL	PHONE_NUM	DOB	DATE_CREATED	NUM_VISIT	RWD_LVL_ID
1	000001	SARAH	RAMIREZ	SRAM@ICLOUD.COM	5204569856	1994-07-15	2018-07-25	000001	N/A
2	000002	ROBERT	SMITH	ROBTHEBOB@GMAIL.COM	4805692488	1972-01-14	2021-11-22	000002	N/A
3	000003	CAROL	REDDING	MISSCAROLR@ICLOUD.COM	4806587215	1969-11-08	2015-12-20	000006	Grammy
4	000004	LUCY	WELLS	WELLSLUCY@ICLOUD.COM	4805462158	2000-12-08	2021-01-20	000017	Oscar
5	000005	YODRI	KIM	YOORIKIM21@ICLOUD.COM	5201469853	1998-04-08	2021-03-24	000005	Grammy
6	000006	ANDREW	LEE	DREWLEE24@ICLOUD.COM	5206350548	1989-09-18	2019-03-15	000031	Tony
7	000007	SEAN	PARKER	PARKINGSEAN@ICLOUD.COM	5201469863	1997-08-25	2020-11-06	000016	Oscar
8	000008	KARLY	JACKSON	KJG@GMAIL.COM	5206978245	1994-08-08	2020-09-24	000001	N/A
9	000009	CARL	RED	CARLR456@ICLOUD.COM	4806594706	1967-11-15	2021-12-28	000003	N/A
10	000010	ADAM	RUIZ	RUIZZAD@GMAIL.COM	6953214562	1981-04-14	2020-01-02	000015	Oscar
11	000012	LUIS	CERVANTEZ	LUISCERV@ICLOUD.COM	5201469875	1995-05-18	2021-01-20	5	Grammy
12	000013	MAX	KITCHEN	KITCHENMAX@GMAIL.COM	5623148965	1999-11-14	2019-10-02	000015	Oscar
13	000014	CELINE	ARIUS	CIACIA@ICLOUD.COM	5204816985	2000-12-18	2021-01-10	000005	Grammy
14	000015	ANDRES	LOPEZ	DRESDRS@ICLOUD.COM	5204163259	2001-09-13	2020-04-19	000006	Grammy
15	000016	LUIS	AVILA	VILAVILA@ICLOUD.COM	2510365942	2001-08-24	2021-11-14	000003	N/A
16	000017	SIRI	PARKS	PARKSIRI@ICLOUD.COM	5201459863	1998-09-25	2020-12-06	000003	N/A
17	000018	SEANS	PITT	PITTS@ICLOUD.COM	5401659853	1995-05-15	2021-11-16	000003	N/A
18	000019	KARLA	JACKETT	KJJ@GMAIL.COM	5207858685	1994-11-27	2020-09-25	000003	N/A
19	000020	DANIELA	RUIZ	RUIDANIELA@GMAIL.COM	5203214562	1979-06-21	2021-01-02	000015	Oscar
20	000021	DANNY	RUIZ	SICKODAN@GMAIL.COM	5204689523	2001-04-01	2020-05-02	000015	Oscar
21	000022	MARCO	CERVANTEZ	MARCOCEERV@ICLOUD.COM	5205429875	1996-07-18	2020-01-20	000004	N/A
22	000023	ANDREW	GARCIA	DREWBOO@ICLOUD.COM	5204538548	1985-09-25	2020-04-17	000006	Grammy
23	000024	ANDREA	AVII A	DRFAAVII A@ICLOUD.COM	52011365942	2001-11-27	2020-11-14	000003	N/A

Query executed successfully. | wpcmssql.cloud.wpcarey.asu... | ASUAD\jecasil1 (118) | BonlineB360 | 00:00:00 | 57 rows

Customer Table Figure (Figure 4.3)

SQLQuery3.sql - wp...UAD\jecasil1 (118)\* -> SQLQuery2.sql - wp...UAD\jecasil1 (111)\* -> SQLQuery1.sql - wp...UAD\jecasil1 (106)\*

Select \*  
From CUSTOMER

Results Messages

CUSTOMER_ID	F_NAME	L_NAME	EMAIL	PHONE_NUM	DOB	DATE_CREATED	NUM_VISIT	RWD_LVL_ID	
21	000022	MARCO	CERVANTEZ	MARCOCERV@ICLOUD.COM	5205429875	1996-07-18	2020-01-20	000004	N/A
22	000023	ANDREW	GARCIA	DREWBOO@ICLOUD.COM	5204538548	1985-09-25	2020-04-17	000006	Grammy
23	000024	ANDREA	AVILA	DREAVILA@ICLOUD.COM	520365942	2001-11-27	2020-11-14	000003	N/A
24	000025	CELINA	SANCHEZ	CIASANCHE@ICLOUD.COM	5207896985	1998-04-21	2021-10-15	000005	Grammy
25	000026	VICTOR	LOPEZ	VICYVIC@ICLOUD.COM	5203332359	2002-09-20	2021-04-19	000006	Grammy
26	000027	ADAMINA	LOPEZ	LOVEMINA@ICLOUD.COM	5203163456	2001-10-27	2021-04-24	000006	Grammy
27	000028	LOLA	LOPEZ	LOLOLA@ICLOUD.COM	4804163259	2000-12-18	2021-07-19	000006	Grammy
28	000029	MADDY	WELLS	MADS@ICLOUD.COM	8605462158	2001-12-08	2018-01-20	000005	Grammy
29	000030	SUMMER	WATERS	SUMMERA@ICLOUD.COM	5205482158	2001-11-08	2021-04-20	000005	Grammy
30	000031	BRAND...	SMITH	BRANDONBOO@GMAIL.COM	480562488	1976-01-17	2017-11-22	000015	Oscar
31	000032	ROBERT	SMART	ROBSMART@GMAIL.COM	4875692488	1978-01-14	2020-12-22	000015	Oscar
32	000033	ZABDIEL	SANTOS	ZABI@GMAIL.COM	6205692488	1970-10-14	2021-10-02	000015	Oscar
33	000034	ANDREA	RAMOS	RAMOSA@ICLOUD.COM	5201365942	2000-12-27	2019-11-14	000003	N/A
34	000035	MANUEL	AVILA	MANNYA@ICLOUD.COM	5275365942	2000-08-27	2020-04-14	000003	N/A
35	000036	SARAH	VALDEZ	SRAVV@ICLOUD.COM	4804569856	1996-07-25	2017-08-27	000009	Grammy
36	000037	BRANDIE	RED	BRAIDRED@ICLOUD.COM	4806474706	1995-08-15	2021-07-28	000004	N/A
37	000038	CARL	ROBERTS	CARLROBBY@ICLOUD.COM	4806794706	1969-07-17	2018-01-28	000004	N/A
38	000039	CARLOS	CISNEROS	CARLOSS@ICLOUD.COM	5206994706	1989-04-01	2020-05-28	000004	N/A
39	000040	CAROL...	FUENTES	CAROCARO@ICLOUD.COM	8966594706	1995-04-04	2020-02-22	000004	N/A
40	000041	KIM	ALINA	AKIM@ICLOUD.COM	5201236942	2001-10-27	2021-11-04	000003	N/A
41	000042	MINHO	LEE	LMIN@GMAIL.COM	5204698524	1987-06-22	2020-02-24	000004	N/A
42	000043	ASHLEY	JOHNSON	ASH@GMAIL.COM	5806567848	1984-03-14	2020-05-26	000004	N/A
43	000060	EIJAH	JACKSON	EJAC@ICLOUD.COM	2096845149	2001-12-05	2022-04-16	000003	N/A

Query executed successfully. wpcmssql.cloud.wpcarey.asu... ASUAD\jecasil1 (118) BonlineB360 00:00:00 57 rows

Customer Table (continued)

SQLQuery3.sql - wp...UAD\jecasil1 (118)\* -> SQLQuery2.sql - wp...UAD\jecasil1 (111)\* -> SQLQuery1.sql - wp...UAD\jecasil1 (106)\*

Select \*  
From CUSTOMER

Results Messages

CUSTOMER_ID	F_NAME	L_NAME	EMAIL	PHONE_NUM	DOB	DATE_CREATED	NUM_VISIT	RWD_LVL_ID	
36	000037	BRANDIE	RED	BRAIDRED@ICLOUD.COM	4806474706	1995-08-15	2021-07-28	000004	N/A
37	000038	CARL	ROBERTS	CARLROBBY@ICLOUD.COM	4806794706	1969-07-17	2018-01-28	000004	N/A
38	000039	CARLOS	CISNEROS	CARLOSS@ICLOUD.COM	5206994706	1999-04-01	2020-05-28	000004	N/A
39	000040	CAROL...	FUENTES	CAROCARO@ICLOUD.COM	8966594706	1995-04-04	2020-02-22	000004	N/A
40	000041	KIM	ALINA	AKIM@ICLOUD.COM	5201236942	2001-10-27	2021-11-04	000003	N/A
41	000042	MINHO	LEE	LMIN@GMAIL.COM	5204698524	1987-06-22	2020-02-24	000004	N/A
42	000043	ASHLEY	JOHNSON	ASH@GMAIL.COM	5806567848	1984-03-14	2020-05-26	000004	N/A
43	000060	EIJAH	JACKSON	EJAC@ICLOUD.COM	2096845149	2001-12-05	2022-04-16	000003	N/A
44	000061	JOHN	SNOW	JOHN14@GMAIL.COM	6574536485	2001-04-04	2020-05-12	000003	N/A
45	100066	Bat	Man	TheBatman@gmail.com	2098709560	1989-04-20	2022-04-22	000003	N/A
46	100067	Cat	Woman	TheCatWoman@gmail.com	2099939895	1989-05-20	2022-04-22	000003	N/A
47	100076	Bat	Man	TheBatman@gmail.com	2098709560	1989-04-20	2022-04-22	000003	N/A
48	100077	Cat	Woman	TheCatWoman@gmail.com	2099939895	1989-05-20	2022-04-22	000003	N/A
49	100078	NAMER	ERIC	ENAM@ICLOUD.COM	5206686299	1988-12-17	2022-02-12	000015	Oscar
50	100100	SEOKJIN	KIM	WORLDWIDEHANDSOME@I...	5208709560	1992-12-04	2020-04-22	000003	N/A
51	100101	YOONGI	MIN	AUGUSTD@ICLOUD.COM	5208509560	1993-03-09	2020-04-24	000003	N/A
52	100102	HOSEOK	JUNG	IAMYOURHOPE@ICLOUD.C...	4808709560	1994-02-18	2019-04-22	000003	N/A
53	100103	NAMJO...	KIM	RIM@ICLOUD.COM	5208709560	1994-09-12	2020-09-22	000003	N/A
54	100104	JIMIN	PARK	JIMINJIMOUT@ICLOUD.COM	5204709590	1995-10-13	2022-04-22	000003	N/A
55	100105	TAEHY...	KIM	VBABAY@ICLOUD.COM	5207509560	1995-12-30	2018-04-22	000003	N/A
56	100106	JUNGK...	JEON	KOOKIE@ICLOUD.COM	5206707560	1997-09-01	2020-07-29	000003	N/A
57	200000	TAY	SMITH	TAYSMITH@GMAIL.COM	2124026890	2000-06-12	2022-02-27	000001	N/A

Query executed successfully. wpcmssql.cloud.wpcarey.asu... ASUAD\jecasil1 (118) BonlineB360 00:00:00 57 rows

Customer Table (continued)

SQLQuery5.sql - wp...SUAD\jecasil1 (85)\* | SQLQuery3.sql - wp...UAD\jecasil1 (118)\* | SQLQuery2.sql - wp...UAD\jecasil1 (111)\* | SQLQuery1.sql - wp...UAD\jecasil1 (106)\*

```
select *
From Inventory
```

100 %

Results Messages

	INVENTORY_ID	RESTOCK_LEVEL	ITEM_DESC	SERVING_SIZE	QOH	QOO
1	400	800	BEER	12 OZ	912	500
2	410	800	WINE	5 OZ	789	0
3	420	800	COCKTAIL	1.5 OZ	723	0
4	530	300	VEGGIE STICKS W/ DRESSING	NULL	459	0
5	531	300	VEGGIE STICKS W/ HUMMUS	NULL	250	0
6	532	50	SIDE OF DRESSING	NULL	42	0
7	533	50	SIDE OF HUMMUS	NULL	31	100
8	550	300	SOFT PRETZEL BITES W/ CHEESE	NULL	460	0
9	551	300	SOFT PRETZEL BITES W/ MUSTARD	NULL	397	0
10	552	50	SIDE OF CHEESE	NULL	22	0
11	553	50	SIDE OF MUSTARD	NULL	11	100
12	580	300	SIDE OF FIRES	NULL	387	0
13	581	200	SIDE OF ONION RINGS	NULL	99	200
14	582	100	SIDE SALAD	NULL	129	0
15	590	500	HAMBURGER W/ CHEESE	NULL	99	0
16	595	250	VEGGIE BURGER W/ CHEESE	NULL	247	0
17	600	500	M&MS	NULL	41	0
18	601	500	PEANUT M&MS	NULL	232	100
19	610	300	SKITTLES	NULL	460	0
20	620	300	REDVINES	NULL	500	0
21	630	200	JUNIOR MINTS	NULL	392	1000
22	640	100	GOOD & PLENTY	NULL	39	100
23	700	100	8 INCH CHEESE PIZZA	8 IN	59	150
24	701	50	10 INCH CHEESE PIZZA	10 IN	45	0

Query executed successfully. | wpcmssql.cloud.wpcarey.asu.... | ASUAD\jecasil1 (85) | BonlineB360 | 00:00:00 | 42 rows

Inventory Table (Figure 4.4)

SQLQuery5.sql - wp...SUAD\jecasil1 (85)\* | SQLQuery3.sql - wp...UAD\jecasil1 (118)\* | SQLQuery2.sql - wp...UAD\jecasil1 (111)\* | SQLQuery1.sql - wp...UAD\jecasil1 (106)\*

```
select *
From Inventory
```

100 %

Results Messages

	INVENTORY_ID	RESTOCK_LEVEL	ITEM_DESC	SERVING_SIZE	QOH	QOO
20	620	300	REDVINES	NULL	500	0
21	630	200	JUNIOR MINTS	NULL	392	1000
22	640	100	GOOD & PLENTY	NULL	39	100
23	700	100	8 INCH CHEESE PIZZA	8 IN	59	150
24	701	50	10 INCH CHEESE PIZZA	10 IN	45	0
25	702	100	14 INCH CHEESE PIZZA	14 IN	20	200
26	710	100	8 INCH VEGGIE PIZZA	8 IN	5	150
27	711	50	10 INCH VEGGIE PIZZA	10 IN	59	0
28	712	100	14 INCH VEGGIE PIZZA	14 IN	130	200
29	720	100	8 INCH PEPPERONI PIZZA	8 IN	120	250
30	721	50	10 INCH PEPPERONI PIZZA	10 IN	90	100
31	722	100	14 INCH PEPPERONI PIZZA	14 IN	119	0
32	730	100	8 INCH MEAT LOVERS PIZZA	8 IN	100	0
33	731	50	10 INCH MEAT LOVERS PIZZA	10 IN	75	0
34	732	100	14 INCH MEAT LOVERS PIZZA	14 IN	119	0
35	800	700	SMALL POPCORN	SMALL	687	0
36	801	700	MEDIUM POPCORN	MED	688	0
37	802	700	LARGE POPCORN	LARG	688	0
38	803	700	EXTRA LARGE POPCORN	XL	686	0
39	900	500	SMALL SODA	SMALL	411	0
40	901	500	MEDIUM SODA	MED	500	0
41	903	500	LARGE SODA	LARG	919	0
42	904	500	EXTRA LARGE SODA	XL	878	0

Query executed successfully. | wpcmssql.cloud.wpcarey.asu.... | ASUAD\jecasil1 (85) | BonlineB360 | 00:00:00 | 42 rows

Inventory Table (continued)

SQLQuery6.sql - wp...SUAD\jecasil1 (85)\*

Select \*

100 %

Results Messages

	SCREEN_ID	CAPACITY	THEATRE_ID
1	101	250	101
2	102	250	101
3	103	250	101
4	104	250	109
5	105	250	109
6	106	250	108
7	107	250	108
8	108	250	108
9	109	250	107
10	110	250	107
11	111	250	107
12	112	250	106
13	113	250	106
14	114	250	106
15	115	250	105
16	116	250	105
17	117	250	105
18	118	250	104
19	119	250	104
20	120	250	104
21	121	250	102
22	122	250	102
23	123	250	102
24	124	250	103

Query executed successfully.

wpcmssql.cloud.wpcarey.asu.... | ASUAD\jecasil1 (85) | BonlineB360 | 00:00:00 | 40 rows

Screen Table (Figure 4.5)

SQLQuery6.sql - wp...SUAD\jecasil1 (85)\*

Select \*

100 %

Results Messages

	SCREEN_ID	CAPACITY	THEATRE_ID
18	118	250	104
19	119	250	104
20	120	250	104
21	121	250	102
22	122	250	102
23	123	250	102
24	124	250	103
25	125	250	103
26	126	250	103
27	127	250	109
28	128	250	110
29	129	250	110
30	130	250	110
31	131	250	111
32	132	250	111
33	133	250	111
34	134	250	112
35	135	250	112
36	136	250	112
37	137	250	112
38	138	250	111
39	139	250	110
40	140	250	109

Query executed successfully.

wpcmssql.cloud.wpcarey.asu.... | ASUAD\jecasil1 (85) | BonlineB360 | 00:00:00 | 40 rows

Screen Table (continued)

SQLQuery7.sql - wp...SUAD\jecasil1 (88)\* × SQLQuery6.sql - wp...SUAD\jecasil1 (85)\*

```
Select * From MOVIE
```

Results Messages

	MOVIE_ID	TITLE	RATING	RELEASE_YR
1	A10001	HARRY POTTER & THE SORCERERS STONE	PG	2001-01-01
2	A10002	TITANIC	PG-13	1997-01-01
3	A10003	STAR WARS EPISODE IX	PG-13	2019-01-01
4	A10004	BACK TO THE FUTURE	PG	1985-01-01
5	A10005	JAWS	PG	1975-01-01
6	A10006	JURASSIC PARK	PG-13	1993-01-01
7	A10007	GHOST	PG-13	1990-01-01
8	A10008	LION KING	G	1994-01-01
9	A10009	TOY STORY	G	1995-01-01
10	A10010	THE AVENGERS	PG-13	2012-01-01
11	A10011	GREASE	PG	1978-01-01
12	A10012	ROCKY	PG	1976-01-01
13	A10013	HARRY POTTER & CHAMBER OF SECRETS	PG	2002-01-01
14	A10014	HARRY POTTER & PRISONER OF AZKABAN	PG	2004-01-01
15	A10015	HARRY POTTER & GOBLET OF FIRE	PG-13	2005-01-01
16	A10016	HARRY POTTER & ORDER OF THE PHOENIX	PG-13	2007-01-01
17	A10017	HARRY POTTER & HALF BLOOD PRINCE	PG	2009-01-01
18	A10018	HARRY POTTER & DEATHLY HALLOWS P1	PG-13	2010-01-01
19	A10019	HARRY POTTER & DEATHLY HALLOWS P2	PG-13	2011-01-01
20	A10020	STAR WARS	PG	1977-01-01
21	A10021	STAR WARS EPISODE V	PG	1980-01-01
22	A10022	STAR WARS EPISODE VI	PG-13	1983-01-01
23	A10023	STAR WARS EPISODE I	PG	1999-01-01

Query executed successfully. | wpcmssql.cloud.wpcarey.asu.... | ASUAD\jecasil1 (88) | BonlineB360 | 00:00:00 | 42 rows

Movie Table (Figure 4.6)

SQLQuery7.sql - wp...SUAD\jecasil1 (88)\* × SQLQuery6.sql - wp...SUAD\jecasil1 (85)\*

```
Select * From MOVIE
```

Results Messages

	MOVIE_ID	TITLE	RATING	RELEASE_YR
21	A10021	STAR WARS EPISODE V	PG	1980-01-01
22	A10022	STAR WARS EPISODE VI	PG-13	1983-01-01
23	A10023	STAR WARS EPISODE I	PG	1999-01-01
24	A10024	STAR WARS EPISODE II	PG	2002-01-01
25	A10025	STAR WARS EPISODE III	PG	2005-01-01
26	A10026	STAR WARS EPISODE VII	PG-13	2015-01-01
27	A10027	STAR WARS EPISODE VIII	PG-13	2017-01-01
28	A10028	THE AVENGERS: AGE OF ULTRON	PG-13	2015-01-01
29	A10029	TOY STORY 2	G	1999-01-01
30	A10030	TOY STORY 3	G	2010-01-01
31	A10031	TOY STORY 4	G	2019-01-01
32	A10032	THE AVENGERS: INFINITY WAR	PG-13	2018-01-01
33	A10033	THE AVENGERS: END GAME	PG-13	2019-01-01
34	A10034	BACK TO THE FUTURE P2	PG	1989-01-01
35	A10035	BACK TO THE FUTURE P3	PG	1990-01-01
36	A10036	KILL BILL: VOL 1	R	2003-01-01
37	A10037	KILL BILL: VOL 2	R	2004-01-01
38	A10038	INGLORIOUS BASTERDS	R	2009-01-01
39	A10039	SLEEPING BEAUTY	G	1959-01-01
40	A10040	SNOW WHITE & THE SEVEN DWARVES	G	1937-01-01
41	A10041	UP	PG	2009-01-01
42	A10042	DESPICABLE ME	PG	2010-01-01

Query executed successfully. | wpcmssql.cloud.wpcarey.asu.... | ASUAD\jecasil1 (88) | BonlineB360 | 00:00:00 | 42 rows

Movie Table (continued)

SQLQuery8.sql - wp...SUAD\jecasil1 (51)\* × SQLQuery7.sql - wp...SUAD\jecasil1 (88)\* SQLQuery6.sql - wp...SUAD\jecasil1 (85)\*

```
Select *
From DATI
```

100 %

Results Messages

	DATE_TIME_ID	SCREEN_ID	MOVIE_ID	TICKET_DATE
1	20110101121111	128	A10028	2021-10-21
2	20110304040454	129	A10029	2020-03-31
3	20110304050607	106	A10006	2021-08-19
4	20110405021055	107	A10007	2022-01-22
5	20110405040406	112	A10012	2021-05-08
6	20110504040550	138	A10038	2020-08-09
7	20110505121234	133	A10033	2022-02-28
8	20110505125505	140	A10040	2021-04-14
9	20110606125505	110	A10010	2021-06-09
10	20110607115959	108	A10008	2022-05-01
11	20110825034511	104	A10004	2022-05-01
12	20110908034554	119	A10019	2022-07-04
13	20111111111111	125	A10025	2021-05-04
14	20111223020202	117	A10017	2022-03-03
15	20111231090959	135	A10035	2020-09-10
16	2012010315425	109	A10009	2021-01-12
17	20120111011111	126	A10026	2022-02-22
18	20120202020202	111	A10011	2020-03-02
19	20120203044554	136	A10036	2021-04-20
20	20120402010159	123	A10023	2022-05-19
21	20120404113223	118	A10018	2021-04-29
22	20120405010220	130	A10030	2022-04-30
23	20120405104509	105	A10005	2020-03-07
24	20120405122424	134	A10034	2021-09-30
25	20120506073434	113	A10013	2021-10-31
26	20120807090859	120	A10020	2021-11-27
27	20120807115904	132	A10032	2021-12-12
28	20120810095454	139	A10039	2021-12-01
29	20120909045454	122	A10022	2021-08-08
30	20121010101010	115	A10015	2021-07-13
31	20121019101403	101	A10001	2022-06-20
32	20121119081405	102	A10002	2022-03-05
33	20121203124503	116	A10016	2022-04-15
34	20121212121212	124	A10024	2022-02-14
35	20121230121211	127	A10027	2021-08-23
36	20130101010101	114	A10014	2021-07-14
37	20130101032121	131	A10031	2021-05-19
38	20131225073055	103	A10003	2021-06-12

Query executed successfully.

wpcmssql.cloud.wpcarey.asu... | ASUAD\jecasil1 (51) | BonlineB360 | 00:00:00 | 38 rows

Date and Time Table (Figure 4.7)

SQLQuery8.sql - wp...SUAD\jecasil1 (51)\* × SQLQuery7.sql - wp...SUAD\jecasil1 (88)\* SQLQuery6.sql - wp...SUAD\jecasil1 (85)\*

```
Select *
From DATI
```

100 %

Results Messages

	DATE_TIME_ID	SCREEN_ID	MOVIE_ID	TICKET_DATE
17	20120111011111	126	A10026	2022-02-22
18	20120202020202	111	A10011	2020-03-02
19	20120203044554	136	A10036	2021-04-20
20	20120402010159	123	A10023	2022-05-19
21	20120404113223	118	A10018	2021-04-29
22	20120405010220	130	A10030	2022-04-30
23	20120405104509	105	A10005	2020-03-07
24	20120405122424	134	A10034	2021-09-30
25	20120506073434	113	A10013	2021-10-31
26	20120807090859	120	A10020	2021-11-27
27	20120807115904	132	A10032	2021-12-12
28	20120810095454	139	A10039	2021-12-01
29	20120909045454	122	A10022	2021-08-08
30	20121010101010	115	A10015	2021-07-13
31	20121019101403	101	A10001	2022-06-20
32	20121119081405	102	A10002	2022-03-05
33	20121203124503	116	A10016	2022-04-15
34	20121212121212	124	A10024	2022-02-14
35	20121230121211	127	A10027	2021-08-23
36	20130101010101	114	A10014	2021-07-14
37	20130101032121	131	A10031	2021-05-19
38	20131225073055	103	A10003	2021-06-12

Query executed successfully.

wpcmssql.cloud.wpcarey.asu... | ASUAD\jecasil1 (51) | BonlineB360 | 00:00:00 | 38 rows

Date and Time Table (continued)

SQLQuery9.sql - wp...SUAD\jecasil1 (91)\* | SQLQuery8.sql - wp...SUAD\jecasil1 (51)\* | SQLQuery7.sql - wp...SUAD\jecasil1 (88)\* | SQLQuery6.sql - wp...SUAD\jecasil1 (85)\*

```
Select * From TICKET
```

100 %

Results Messages

	TICKET_ID	MOVIE_ID	CUSTOMER_ID	DATE_TIME_ID	SEAT
1	100001	A10001	000001	20121019101403	900001
2	100002	A10002	000002	20121019101403	900001
3	100003	A10003	000003	20131225073055	900003
4	100004	A10004	000004	20110825034511	900004
5	100005	A10005	000005	20120405104509	900005
6	100006	A10006	000006	20110304050607	900006
7	100007	A10007	000007	20110405021059	900007
8	100008	A10008	000008	20110607115959	900008
9	100009	A10009	000009	20120103115425	900009
10	100010	A10010	000010	20110606125505	900010
11	100011	A10001	000001	20121019101403	900001
12	100012	A10012	000012	20110405040406	900012
13	100013	A10013	000013	20120506073434	900013
14	100014	A10014	000014	20130101010101	900014
15	100015	A10015	000015	20121010101010	900015
16	100016	A10016	000016	20121203124503	900016
17	100017	A10017	000017	20111223020202	900017
18	100018	A10018	000018	20120404113223	900018
19	100019	A10019	000019	20110908034554	900019
20	100020	A10020	000020	20120807090859	900020
21	100021	A10002	000002	20121019101403	900001
22	100022	A10022	000022	20120909045454	900022
23	100023	A10023	000023	20120402010159	900023

Query executed successfully. | wpcmssql.cloud.wpcarey.asu.... | ASUAD\jecasil1 (91) | BonlineB360 | 00:00:00 | 46 rows

Ticket Table (Figure 4.8)

SQLQuery9.sql - wp...SUAD\jecasil1 (91)\* | SQLQuery8.sql - wp...SUAD\jecasil1 (51)\* | SQLQuery7.sql - wp...SUAD\jecasil1 (88)\* | SQLQuery6.sql - wp...SUAD\jecasil1 (85)\*

```
Select * From TICKET
```

100 %

Results Messages

	TICKET_ID	MOVIE_ID	CUSTOMER_ID	DATE_TIME_ID	SEAT
23	100023	A10023	000023	20120402010159	900023
24	100024	A10024	000024	20121212121212	900024
25	100025	A10025	000025	20111111111111	900025
26	100026	A10026	000026	20121110111111	900026
27	100027	A10027	000027	20121230121211	900027
28	100028	A10028	000028	20110101121111	900028
29	100029	A10029	000029	20110304040454	900029
30	100030	A10030	000030	20120405010228	900030
31	100031	A10031	000031	20130101032121	900031
32	100032	A10032	000032	20120807115904	900032
33	100033	A10033	000033	20110505121234	900033
34	100034	A10034	000034	20120405122424	900034
35	100035	A10035	000035	20111231090959	900035
36	100036	A10036	000036	20120203044554	900036
37	100037	A10002	000002	20121019101403	900001
38	100038	A10038	000038	20110504040550	900038
39	100039	A10039	000039	20120810095454	900039
40	100040	A10040	000040	20110505125505	900040
41	100041	A10027	100102	20121230121211	900025
42	100042	A10027	100103	20121230121211	900026
43	100043	A10027	100104	20121230121211	900027
44	100044	A10018	000012	20120404113223	900028
45	100045	A10018	000012	20120404113223	900028

Query executed successfully. | wpcmssql.cloud.wpcarey.asu.... | ASUAD\jecasil1 (91) | BonlineB360 | 00:00:00 | 46 rows

Ticket Table (continued)

SQLQuery9.sql - wp...SUAD\jecasil1 (91)\* | SQLQuery8.sql - wp...SUAD\jecasil1 (51)\* | SQLQuery7.sql - wp...SUAD\jecasil1 (88)\* | SQLQuery6.sql - wp...SUAD\jecasil1 (85)\*

```
Select *
From TICKET
```

100 %

Results Messages

TICKET_ID	MOVIE_ID	CUSTOMER_ID	DATE_TIME_ID	SEAT	
25	A10025	000025	2011111111111	900025	
26	A10026	000026	2012011101111	900026	
27	A10027	000027	20121230121211	900027	
28	A10028	000028	20110101121111	900028	
29	A10029	000029	20110304040454	900029	
30	A10030	000030	20120405010220	900030	
31	A10031	000031	20130101032121	900031	
32	A10032	000032	20120807115904	900032	
33	A10033	000033	20110505121234	900033	
34	A10034	000034	20120405122424	900034	
35	A10035	000035	20111231090956	900035	
36	A10036	000036	20120203044554	900036	
37	A10037	000002	20121019101403	900001	
38	A10038	000038	20110504040550	900038	
39	A10039	000039	20120810095454	900039	
40	A10040	000040	20110505125505	900040	
41	A10041	A10027	100102	20121230121211	900025
42	A10042	A10027	100103	20121230121211	900026
43	A10043	A10027	100104	20121230121211	900027
44	A10044	A10018	000012	20120404113223	900028
45	A10045	A10018	000012	20120404113223	900028
46	A10047	A10018	000012	20120404113223	900028

Query executed successfully.

wpcmssql.cloud.wpcarey.asu.... ASUAD\jecasil1 (91) BonlineB360 00:00:00 46 rows

Ticket Table (continued)

SQLQuery10.sql - wp...SUAD\jecasil1 (51)\* | SQLQuery9.sql - wp...SUAD\jecasil1 (91)\*

```
Select *
From Menu
```

100 %

Results Messages

MENU_ID	INVENTORY_ID	QUANTITY	COST	EXT_COST
1	600001	800	2	6.00 12.00
2	600002	801	1	7.00 7.00
3	600003	802	1	8.00 8.00
4	600004	803	3	9.00 27.00
5	600005	900	1	5.50 5.50
6	600006	901	1	6.00 6.00
7	600008	903	1	7.00 7.00
8	600009	904	2	7.50 15.00
9	600010	700	1	7.00 7.00
10	600011	701	1	7.50 7.50
11	600012	702	2	8.00 16.00
12	600013	710	1	7.00 7.00
13	600014	711	1	7.50 7.50
14	600015	712	1	8.00 8.00
15	600016	720	3	7.00 21.00
16	600017	721	4	7.50 30.00
17	600018	722	1	8.00 8.00
18	600019	730	1	7.00 7.00
19	600020	731	4	7.50 30.00
20	600021	732	1	8.00 8.00
21	600022	600	3	4.00 12.00
22	600023	610	1	4.00 4.00
23	600024	620	2	4.00 8.00
24	600025	61	1	4.00 4.00

Query executed successfully.

wpcmssql.cloud.wpcarey.asu.... ASUAD\jecasil1 (51) BonlineB360 00:00:00 42 rows

Menu Table (Figure 4.9)

SQLQuery10.sql - w...SUAD\jecasil1 (51)\* × SQLQuery9.sql - wp...SUAD\jecasil1 (91)\*

Select \*  
From Menu

100 %

Results Messages

	MENU_ID	INVENTORY_ID	QUANTITY	COST	EXT_COST
20	600021	732	1	8.00	8.00
21	600022	600	3	4.00	12.00
22	600023	610	1	4.00	4.00
23	600024	620	2	4.00	8.00
24	600025	601	1	4.00	4.00
25	600026	630	1	4.00	4.00
26	600027	640	5	4.00	20.00
27	600028	550	1	5.00	5.00
28	600029	551	3	5.00	15.00
29	600030	552	1	0.50	0.50
30	600031	553	2	0.50	1.00
31	600032	530	1	1.50	1.50
32	600033	531	4	1.50	6.00
33	600034	532	1	0.50	0.50
34	600035	533	2	0.50	1.00
35	600036	400	1	2.00	2.00
36	600037	410	1	5.00	5.00
37	600038	420	1	10.00	10.00
38	600039	580	1	7.00	7.00
39	600040	581	1	7.00	7.00
40	600041	582	1	2.00	2.00
41	600042	590	1	2.00	2.00
42	600043	595	1	2.00	2.00

Query executed successfully. | wpcmssql.cloud.wpcarey.asu.... | ASUAD\jecasil1 (51) | BonlineB360 | 00:00:00 | 42 rows

Menu Table (continued)

SQLQuery11.sql - w...SUAD\jecasil1 (85)\* × SQLQuery10.sql - w...SUAD\jecasil1 (51)\* × SQLQuery9.sql - wp...SUAD\jecasil1 (91)\*

Select \*  
from INVOICE

100 %

Results Messages

	INVOICE_ID	DEPARTMENT	TOTAL	DISCOUNT	TAX	TIP	EMPLOYEE_ID	MENU_ID	INVENTORY_ID	TICKET_ID	INV_DATE
1	200001	SNACK STAND	12.00	0.00	0.72	2.40	000001	600001	800	100001	09/22/2022
2	200002	SNACK STAND	7.00	0.00	0.42	1.40	000002	600002	801	100002	10/04/2020
3	200003	SNACK STAND	8.00	0.00	0.48	1.60	000003	600003	802	100003	11/17/2021
4	200004	SNACK STAND	27.00	2.70	1.62	5.40	000004	600004	803	100004	05/05/2021
5	200005	SNACK STAND	5.50	0.00	0.33	1.10	000005	600005	900	100005	03/02/2020
6	200006	SNACK STAND	6.00	0.00	0.36	1.20	000006	600006	901	100006	01/17/2021
7	200008	SNACK STAND	7.00	0.00	0.42	1.40	000008	600008	903	100008	09/10/2020
8	200009	SNACK STAND	15.00	1.50	0.90	3.00	000009	600009	904	100009	03/18/2022
9	200010	SNACK STAND	7.00	0.00	0.42	1.40	000014	600010	700	100010	08/31/2020
10	200011	SNACK STAND	7.50	0.00	0.34	1.20	000015	600011	701	100011	10/08/2020
11	200012	SNACK STAND	16.00	1.60	0.96	3.20	000016	600012	702	100012	04/25/2021
12	200013	SNACK STAND	7.00	0.00	0.42	1.40	000017	600013	710	100013	07/25/2021
13	200014	SNACK STAND	7.50	0.00	0.34	1.20	000018	600014	711	100014	11/07/2022
14	200015	SNACK STAND	8.00	0.00	0.48	1.60	000019	600015	712	100015	11/14/2022
15	200016	SNACK STAND	21.00	2.10	1.26	4.20	000020	600016	720	100016	11/15/2020
16	200017	SNACK STAND	30.00	3.00	1.80	6.00	000021	600017	721	100017	05/02/2022
17	200018	SNACK STAND	8.00	0.00	0.48	1.60	000022	600018	722	100018	05/25/2021
18	200019	SNACK STAND	7.00	0.00	0.42	1.40	000023	600019	730	100019	08/30/2022
19	200020	SNACK STAND	30.00	3.00	1.80	6.00	000024	600020	731	100020	12/09/2020
20	200021	SNACK STAND	8.00	0.00	0.48	1.60	000025	600021	732	100021	07/23/2021
21	200022	SNACK STAND	12.00	0.00	0.72	2.40	000026	600022	600	100022	06/27/2020
22	200023	SNACK STAND	4.00	0.00	0.24	0.80	000027	600023	610	100023	02/09/2020
23	200024	SNACK STAND	8.00	0.00	0.48	1.60	000028	600024	620	100024	02/22/2021

Query executed successfully. | wpcmssql.cloud.wpcarey.asu.... | ASUAD\jecasil1 (85) | BonlineB360 | 00:00:00 | 45 rows

Invoice Table (Figure 4.10)

SQLQuery11.sql - w...SUAD\jecasil1 (85)\* SQLQuery10.sql - w...SUAD\jecasil1 (51)\* SQLQuery9.sql - wp...SUAD\jecasil1 (91)\*

Select \* from INVOICE

100 %

Results Messages

	INVOICE_ID	DEPARTMENT	TOTAL	DISCOUNT	TAX	Tip	EMPLOYEE_ID	MENU_ID	INVENTORY_ID	TICKET_ID	INV_DATE
23	200024	SNACK STAND	8.00	0.00	0.48	1.60	000028	600024	620	100024	02/22/2021
24	200025	SNACK STAND	4.00	0.00	0.24	0.80	000029	600025	601	100025	06/13/2020
25	200026	SNACK STAND	4.00	0.00	0.24	0.80	000030	600026	630	100026	10/16/2022
26	200027	SNACK STAND	20.00	2.00	1.20	4.00	000031	600027	640	100027	08/23/2021
27	200028	SNACK STAND	5.00	0.00	0.30	1.00	000032	600028	550	100028	09/02/2021
28	200029	SNACK STAND	15.00	1.50	0.90	3.00	000033	600029	551	100029	05/21/2022
29	200030	SNACK STAND	0.50	0.00	0.03	0.00	000034	600030	552	100030	08/18/2020
30	200031	SNACK STAND	1.00	0.00	0.06	0.00	000035	600031	553	100031	07/26/2022
31	200032	SNACK STAND	1.50	0.00	0.06	0.00	000036	600032	530	100032	05/05/2020
32	200033	SNACK STAND	6.00	0.00	0.36	1.20	000037	600033	531	100033	02/16/2022
33	200034	SNACK STAND	0.50	0.00	0.03	0.00	000038	600034	532	100034	06/02/2021
34	200035	SNACK STAND	1.00	0.00	0.06	0.20	000039	600035	533	100035	11/22/2021
35	200036	BAR	2.00	0.00	0.12	0.00	000040	600036	400	NULL	3/13/2022
36	200037	BAR	5.00	0.00	0.30	1.00	000031	600037	410	NULL	10/16/2020
37	200038	BAR	10.00	0.00	0.60	2.00	000032	600038	420	NULL	02/19/2022
38	200039	SNACK STAND	7.00	0.00	0.42	1.40	000041	600039	580	100036	10/17/2022
39	200040	SNACK STAND	7.00	0.00	0.42	1.40	000041	600040	581	100037	10/11/2020
40	200041	SNACK STAND	2.00	0.00	0.12	0.00	000041	600041	582	100038	04/28/2021
41	200042	SNACK STAND	2.00	0.00	0.12	0.00	000041	600042	590	100039	10/05/2020
42	200043	SNACK STAND	2.00	0.00	0.12	0.00	000041	600043	595	100040	06/20/2021
43	200044	SNACK STAND	5.00	0.00	0.30	1.00	000041	600028	550	100041	07/05/2022
44	200045	SNACK STAND	7.00	0.00	0.40	2.00	000041	600019	730	100042	07/05/2020
45	200047	SNACK STAND	13.00	0.00	0.73	3.00	000001	600014	711	100044	03/01/2022

Query executed successfully.

wpcmssql.cloud.wpcarey.asu... ASUAD\jecasil1 (85) BonlineB360 00:00:00 45 rows

## Invoice Table (continued)

# SCREEN CAPTURE OF ACL TESTS AND RESULTS

ACL Test performed by Christine Beck:  
Calculate Palermo's Average Total, Tip, and Discount amount from all 45 invoices.

Figure 5.1

The screenshot displays two side-by-side windows of an ACL test interface. Both windows have tabs for 'INvoice' and 'Statistics'.

**Left Window (INvoice Tab):**

- Header:** As of: 04/26/2022 20:01:42, Command: STATISTICS ON TOTAL TIP DISCOUNT TO SCREEN NUMBER 5, Table: INvoice.
- TOTAL:**

	Number	Total	Average
Range	-	29.50	-
Positive	45	389.00	8.64
Negative	0	0.00	0.00
Zeros	0	-	-
Totals	45	389.00	8.64
Abs Value	-	389.00	-

Highest	Lowest
30.00	0.50
30.00	0.50
27.00	1.00
21.00	1.00
20.00	1.50
- TIP:**

	Number	Total	Average
Range	-	6.00	-
Positive	37	75.90	2.05
Negative	0	0.00	0.00
Zeros	8	-	-
Totals	45	75.90	1.69
Abs Value	-	75.90	-

Highest	Lowest
6.00	0.00
6.00	0.00
5.40	0.00
4.20	0.00
4.00	0.00
- DISCOUNT:**

	Number	Total	Average
Range	-	3.00	-
Positive	8	17.40	2.18
Negative	0	0.00	0.00
Zeros	37	-	-
Totals	45	17.40	0.39
Abs Value	-	17.40	-

Highest	Lowest
3.00	0.00
3.00	0.00
2.70	0.00
2.10	0.00
2.00	0.00

**Bottom Window (Statistics Tab):**

- Tables:** Text
- Records:** 45

Top of Screen Capture of ACL Test Results

Bottom of Screen Capture of ACL Test Results

Based on this test it would appear that, across the 45 Invoices we input into Palermo's new database, that the average Invoice Total was \$8.64, with Employees averaging \$1.69 in tips per invoice (about 19.5% of invoice total), and the company discounted a total of \$17.40 which averaged to be \$0.39 per invoice.

ACL Test performed by Elijah Jackson:  
Find which inventory items have Quantity on Hand (QOH) is  $\geq 50$

Figure 5.2  
Data results before filter:

INVENTORY_ID	ITEM_DESC	QOH	QOO	RESTOCK_LEVEL	SERVING_SIZE
1 553	SIDE OF MUSTARD	11	100	50	
2 552	SIDE OF CHEESE	22	0	50	
3 601	PEANUT M&MS	232	100	500	
4 595	VEGGIE BURGER W/ CHEESE	247	0	250	
5 531	VEGGIE STICKS W/ HUMMUS	250	0	300	
6 533	SIDE OF HUMMUS	31	100	50	
7 600	M&MS	41	0	500	
8 900	SMALL SODA	411	0	500	SMALL
9 532	SIDE OF DRESSING	42	0	50	
10 701	10 INCH CHEESE PIZZA	45	0	50	10 IN
11 803	EXTRA LARGE POPCORN	686	0	700	XL
12 800	SMALL POPCORN	687	0	700	SMALL
13 802	LARGE POPCORN	688	0	700	LARG
14 801	MEDIUM POPCORN	688	0	700	MED
15 420	COCKTAIL	723	0	800	1.5 OZ
16 410	WINE	789	0	800	5 OZ

Data results after filter:

INVENTORY_ID	ITEM_DESC	QOH	QOO	RESTOCK_LEVEL	SERVING_SIZE
11 803	EXTRA LARGE POPCORN	686	0	700	XL
12 800	SMALL POPCORN	687	0	700	SMALL
13 802	LARGE POPCORN	688	0	700	LARG
14 801	MEDIUM POPCORN	688	0	700	MED
15 420	COCKTAIL	723	0	800	1.5 OZ
16 410	WINE	789	0	800	5 OZ

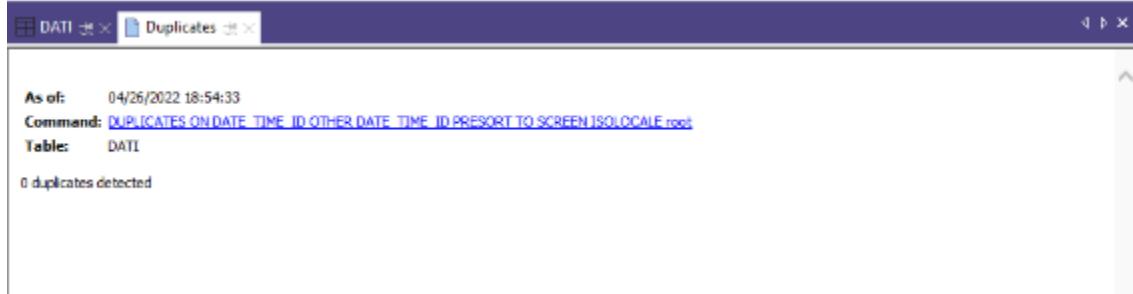
Count number of items where QOH  $\geq 50$

As of:	04/27/2022 10:35:34
Command:	COUNT IF QOH >= '50'
Table:	InventoryJoin
Condition:	QOH >= '50' (6 records matched)
6 records counted	

The count function verifies that there are 6 inventory items that currently have more than 50 on hand.

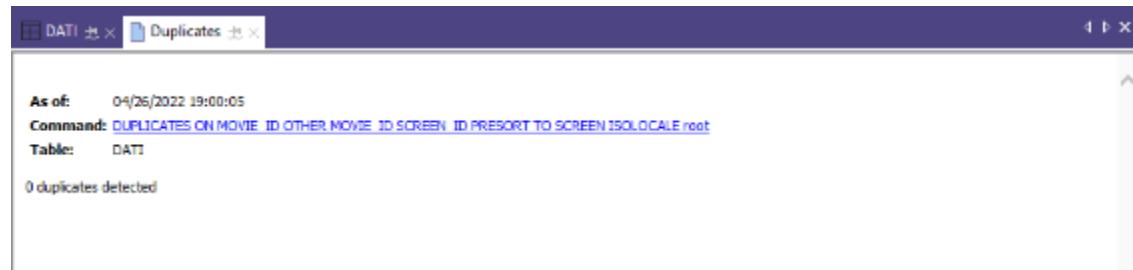
ACL Test performed by Erika Moss:

Figure 5.3



This test was used to determine if there were duplicate screening times. This test shows that there were not duplicate screening times.

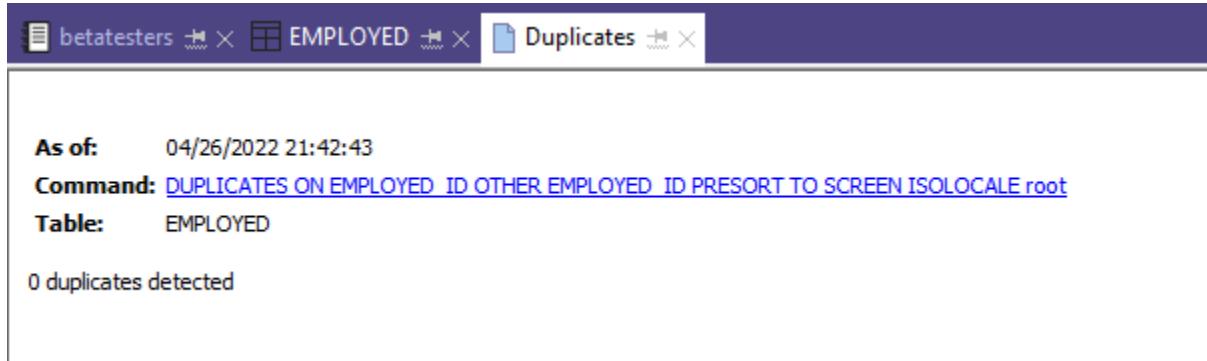
Figure 5.4



This test was performed to check for duplicate movies on the same screen. If there were duplicate movies, a test was needed to make sure that the duplicate movies on the same screen were being played at different theaters. The test found that there were no duplicate movies on the same screens.

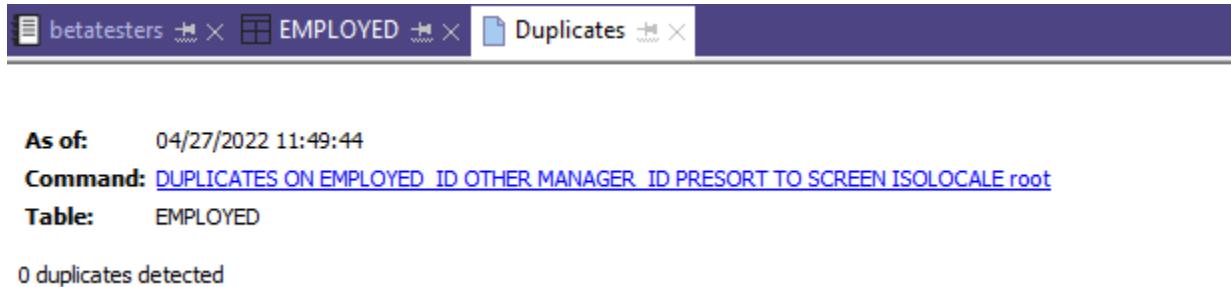
ACL Test performed by Celeste Garcia:  
Check there are no duplicate employees in the system and one manager is assigned per employee.

Figure 5.4



This duplicate test in ACL reveals that there are no duplicate employees added into the system.

Figure 5.5



This duplicate test in ACL revealed that there were no duplicates detected on the Manager ID and Employee ID.

ACL Test performed by Alec Valencia:  
Test if there are any Gaps in Invoice/ticket

Figure 5.6

**As of:** 04/26/2022 16:33:30  
**Command:** [GAPS ON INVOICE\\_ID PRESORT TO SCREEN](#)  
**Table:** Invoice

2 gap ranges detected  
 2 missing items

### Gaps Found Between:

Gap Start (Exclusive)	Gap End (Exclusive)	Number of Missing Items
200,006	200,008	1
200,045	200,047	1

**As of:** 04/26/2022 16:32:44  
**Command:** [GAPS ON TICKET\\_ID PRESORT TO SCREEN](#)  
**Table:** Ticket

1 gap ranges detected  
 1 missing items

### Gaps Found Between:

Gap Start (Exclusive)	Gap End (Exclusive)	Number of Missing Items
100,045	100,047	1

This test showed there were 2 Gaps found in Invoice and 1 Gap found in Ticket.

ACL Test performed by Ryu Lewis

Check for duplicate customers

Figure 5.7

The screenshot shows a terminal window with the following text:

```
As of: 04/28/2022 07:47:14
Command: DUPLICATES ON CUSTOMER_ID OTHER CUSTOMER_ID DATE_CREATED DOB EMAIL_F_NAME L_NAME NUM_VISIT PHONE_NUM RWD_LVL_ID PRESORT_TO SCREEN_ISOLOCALE root
Table: TeamBCustomer
0 duplicates detected
```

There are no duplicate customers

ACL Test Performed by Jorge Castro

Count how many tickets were assigned to a customer or not Figure 5.8

**As of:** 04/30/2022 04:53:43  
**Command:** COUNT IF TICKET\_ID >= "0" AND CUSTOMER\_ID >= "0"  
**Table:** TicketB360  
**Condition:** TICKET\_ID >= "0" AND CUSTOMER\_ID >= "0" (46 records matched)

46 records counted



# TECHNICAL ISSUES, QUIRKS, CHALLENGES, LESSONS LEARNED

The Beta Testers have learned many lessons from the process of creating a database. They have learned to have patience in learning SQL, that it's alright to restart in order to improve the project, and that everything eventually falls into place.

The Beta Testers experienced a handful of issues with trying to get everything into SQL. The coding for tables or data inserts would give them problems. They had to learn to have patience with themselves and each other as everyone learned SQL. Through a lot of trial and error, they were able to work through the problems and get the database built.

There were a few times when sections of the project had to be restarted so they could be improved. The SQL tables had to be taken out and put back together once significant changes were made to reflect the changes to the ERD. Putting the tables back in again also made it easier to make edits outside of SQL and then put the finished product into the database.

The Beta Testers learned that through hard work, everything eventually comes together. With patience, the database came together to be a beautiful project to present to Palermo.

The Beta Testers ran into a few technical issues during the completion of this project. Amazon Workspaces stopped working for hours at a time and the group members could not log in. This made it hard to complete group tasks before group meetings. The members had to continue trying to login or restart the program multiple times before it would open.

The link between Access and SQL did not always function properly. The link had to be recreated to get Access and SQL to update each other.

Besides these challenges and technical issues, the team members learned a lot while completing this project. Putting together a database was a new experience, and the group is pleased with the results!

## APPENDICES A: SQL

### ORIGINAL TABLE AND INSERT VALUE SCRIPTS

Customer Table

CREATE TABLE CUSTOMER(		
CUSTOMER_ID	INT IDENTITY(1, 1) PRIMARY KEY,	
F_NAME	VARCHAR(3 5)	NOT NULL,
L_NAME	VARCHAR(3 5)	NOT NULL,
EMAIL	VARCHAR(3 5)	NOT NULL,
PHONE_NUM	CHAR(10)	NOT NULL,
DOB	DATE	NOT NULL,
DATE_CREATED	DATE	NOT NULL,
NUM_VISIT	CHAR(6)	NOT NULL,
RWD_LVL_ID	CHAR(6)	NOT NULL);

INSERT INTO CUSTOMER
VALUES ('000001', 'RAMIREZ', 'SARAH', 'SRAM@ICLOUD.COM', '5204569856', 1994-07-15', '2018-07-25', '000009', 'GRAMMY',)
VALUES ('000002', 'GARCIA', 'SARAH', 'SGARCI@GMAIL.COM', '520', '1999-11-12', '2017-05-28', '000002', 'EMMY')
VALUES ('000003', 'SMITH', 'ROBERT', 'ROBTHEBOB@GMAIL.COM', '4805692488', '1972-01-14', '2021-11-22', '000015', 'OSCAR')
VALUES ('000004', 'REDDING', 'CAROL', 'MISSCAROLR@ICLOUD.COM', '4806587215',

'1969-11-08', '2015-12-20', '000004', 'EMMY')
VALUES ('000005', 'WELLS', 'LUCY', 'WELSLUCY@ICLOUD.COM', '4805462158', '2000-12-08', '2021-01-20', '000005', 'GRAMMY')
VALUES ('000006', 'KIM', 'YOORI', 'YOORIKIM21@ICLOUD.COM', '5201469853', '1998-04-08', '2021-03-24', '000003', 'EMMY')
VALUES ('000007', 'LEE', 'ANDREW', 'DREWLEE24@ICLOUD.COM', '5206350548', '1999-09-18', '2019-03-15', '000006', 'GRAMMY')
VALUES ('000008', 'PARKER', 'SEAN', 'PARKINGSEAN@ICLOUD.COM', '5201469863', '1997-08-25', '2020-11-06', '000003', 'EMMY')
VALUES ('000009', 'JACKSON', 'KARLY', 'KJ@GMAIL.COM', '5206978245', '1994-08-08', '2020-09-24', '000003', 'EMMY')
VALUES ('000010', 'RED', 'CARL', 'CARLR456@ICLOUD.COM', '4806594706', '1967-11-15', '2021-12-28', '000004', 'EMMY')
VALUES ('000011', 'RUIZ', 'ADAM', 'RUIZZAD@GMAIL.COM', '6953214562', '1981-04-14', '2020-01-02', '000015', 'OSCAR')
VALUES ('000012', 'CERVANTEZ', 'LUIS', 'LUISCERV@ICLOUD.COM', '5201469875', '1995-05-18', '2021-01-20', '000004', 'EMMY')
VALUES ('000013', 'KITCHEN', 'MAX', 'KITCHENMAX@GMAIL.COM', '5623148965', '1999-11-14', '2019-10-02', '000015', 'OSCAR')
VALUES ('000001', 'ARIUS', 'CELINE', 'CIACIA@ICLOUD.COM', '5204816985', '2000-12-18', '2021-01-10', '000005', 'GRAMMY')
VALUES ('000001', 'LOPEZ', 'ANDRES', 'DRESDRES@ICLOUD.COM', '5204163259', '2001-09-18', '2020-04-19', '000006', 'GRAMMY')
VALUES ('000001', 'AVILA', 'LUIS', 'VILAVILA@ICLOUD.COM', '2510365942', '2001-08-24', '2021-11-14', '000003', 'EMMY')
VALUES ('000001', 'PARKS', 'SIRI', 'PARKSIRI@ICLOUD.COM', '5201459863', '1998-09-25', '2020-12-06', '000003', 'EMMY')
VALUES ('000001', 'PITT', 'SEANS', 'PITTS@ICLOUD.COM', '5401659853', '1995-05-15', '2021-11-16', '000003', 'EMMY')
VALUES ('000001', 'JACKETT', 'KARLA', 'KJJ@GMAIL.COM', '5207858685', '1994-11-27', '2020-09-25', '000003', 'EMMY')
VALUES ('000001', 'RUIZ', 'DANIELA', 'RUIZDANIELA@GMAIL.COM', '5203214562', '1979-06-21', '2021-01-02', '000015', 'OSCAR')
VALUES ('000001', 'RUIZ', 'DANNY', 'SICKODAN@GMAIL.COM', '5204689523',

'2001-04-01', '2020-15-02', '000015', 'OSCAR')
VALUES ('000001', 'CERVANTEZ', 'MARCO', 'MARCOCERV@ICLOUD.COM', '5205429875', '1996-07-18', '2020-01-20', '000004', 'EMMY')
VALUES ('000001', 'GARCIA', 'ANDREW', 'DREWBOO@ICLOUD.COM', '5204538548', '1985-09-25', '2020-04-17', '000006', 'GRAMMY')
VALUES ('000001', 'AVILA', 'ANDREA', 'DREAAVILA@ICLOUD.COM', '5200365942', '2001-11-27', '2020-11-14', '000003', 'EMMY')
VALUES ('000001', 'SANCHEZ', 'CELINA', 'CIASANCH@ICLOUD.COM', '5207896985', '1998-04-21', '2021-10-15', '000005', 'GRAMMY')
VALUES ('000001', 'LOPEZ', 'VICTOR', 'VICYVIC@ICLOUD.COM', '5203333259', '2002-09-20', '2021-04-19', '000006', 'GRAMMY')
VALUES ('000001', 'LOPEZ', 'ADAMINA', 'LOVEMINA@ICLOUD.COM', '5203163456', '2001-10-27', '2021-04-24', '000006', 'GRAMMY')
VALUES ('000001', 'LOPEZ', 'LOLA', 'LOLOLA@ICLOUD.COM', '4804163259', '2000-12-18', '2021-07-19', '000006', 'GRAMMY')
VALUES ('000001', 'WELLS', 'MADDY', 'MADS@ICLOUD.COM', '8605462158', '2001-12-08', '2018-01-20', '000005', 'GRAMMY')
VALUES ('000001', 'WATERS', 'SUMMER', 'SUMMERA@ICLOUD.COM', '5205482158', '2001-11-08', '2021-04-20', '000005', 'GRAMMY')
VALUES ('000001', 'SMITH', 'BRANDON', 'BRANDONBOO@GMAIL.COM', '4805682488', '1976-01-17', '2017-11-22', '000015', 'OSCAR')
VALUES ('000001', 'SMART', 'ROBERT', 'ROBSMART@GMAIL.COM', '4875692488', '1978-01-14', '2020-12-22', '000015', 'OSCAR')
VALUES ('000001', 'SANTOS', 'ZABDIEL', 'ZABI@GMAIL.COM', '6205692488', '1970-10-14', '2021-10-02', '000015', 'OSCAR')
VALUES ('000001', 'RAMOS', 'ANDREA', 'RAMOSA@ICLOUD.COM', '5201365942', '2000-12-27', '2019-11-14', '000003', 'EMMY')
VALUES ('000001', 'AVILA', 'MANUEL', 'MANNYA@ICLOUD.COM', '5275365942', '2000-08-27', '2020-04-14', '000003', 'EMMY')
VALUES ('000001', 'VALDEZ', 'SARAH', 'SRAVV@ICLOUD.COM', '4804569856', '1996-07-55', '2017-08-27', '000009', 'GRAMMY',)
VALUES ('000001', 'RED', 'BRANDIE', 'BRADRED@ICLOUD.COM', '4806474706', '1995-08-15', '2021-07-28', '000004', 'EMMY')
VALUES ('000001', 'ROBERTS', 'CARL', 'CARLROBBY@ICLOUD.COM', '4806794706',

'1969-07-17', '2018-01-28', '000004', 'EMMY')
VALUES ('000001', 'CINSNEROS', 'CARLOS', 'CARLOSS@ICLOUD.COM', '5206994706', '1999-04-01', '2020-05-28', '000004', 'EMMY')
VALUES ('000001', 'FUENTES', 'CAROLINA', 'CAROCARO@ICLOUD.COM', '8966594706', '1995-04-04', '2020-02-22', '000004', 'EMMY')
VALUES ('000001', 'KIM', 'ALINA', 'AKIM@ICLOUD.COM', '5201236942', '2001-10-27', '2021-11-04', '000003', 'EMMY')

### Theatre Table

CREATE TABLE THEATRE (		
THEATRE_ID	CHAR(6)	NOT NULL,
NAME	VARCHAR(35)	NOT NULL,
ADDRESS	VARCHAR(35)	NOT NULL,
CITY	VARCHAR(35)	NOT NULL,
STATE	CHAR(2)	NOT NULL,
ZIP	CHAR(5)	NOT NULL,
PHN_NUM	CHAR(10)	NOT NULL,
EMAIL	VARCHAR(35)	NOT NULL,
PRIMARY KEY (THEATRE_ID));		

INSERT INTO THEATRE
VALUES('101', 'TITANIC', '1912 S. APRIL AVE.', 'PHOENIX', 'AZ', '85001', '6025557865', 'TITANIC@PALERMO.COM');
INSERT INTO THEATRE
VALUES('102', 'STAR WARS', '52577 N. LUCAS LN.', 'SCOTTSDALE', 'AZ', '85054', '4805555334', 'STARWARS@PALERMO.COM');
INSERT INTO THEATRE
VALUES('103', 'BACK TO THE FUTURE', '85515 W. MCFLY RD.', 'TEMPE', 'AZ', '85202', '6025552225', 'BACK2FUTURE@PALERMO.COM');
INSERT INTO THEATRE

```

VALUES('104', 'JAWS', '62075 S. SPIELBERG ST.', 'FOUNTAIN HILLS', 'AZ', '85268', '4805556722',
'JAWS@PALERMO.COM');

INSERT INTO THEATRE

VALUES('105', 'HARRY POTTER', '111401 W. PRIVET DR.', 'CHANDLER', 'AZ', '85224', '4805554774',
'HARRYPOTTER@PALERMO.COM');

INSERT INTO THEATRE

VALUES('106', 'JURASSIC PARK', '15 W. PREHISTORIC PKWY.', 'MESA', 'AZ', '85201', '4805553466',
'DINOPARK@PALERMO.COM');

INSERT INTO THEATRE

VALUES('107', 'GHOST', '71390 N. SPECTER ST.', 'GLENDALE', 'AZ', '85301', '6235553323',
'GHOST@PALERMO.COM');

INSERT INTO THEATRE

VALUES('108', 'THE LION KING', '1994 E. PRIDE ROCK BLVD.', 'PEORIA', 'AZ', '85301', '6025557865',
'LIONKING@PALERMO.COM');

INSERT INTO THEATRE

VALUES('109', 'TOY STORY', '95113-A S. PIXAR PL.', 'SURPRISE', 'AZ', '85335', '6235552639',
'TOYSTORY@PALERMO.COM');

INSERT INTO THEATRE

VALUES('110', 'THE AVENGERS', '20126 E. ASSEMBLE AVE.', 'QUEEN CREEK', 'AZ', '85140',
'4805557826', 'AVENGERS@PALERMO.COM');

INSERT INTO THEATRE

VALUES('111', 'GREASE', '7858 E. LIGHTENING LN.', 'GOODYEAR', 'AZ', '85326', '6235555683',
'GREASE@PALERMO.COM');

INSERT INTO THEATRE

VALUES('112', 'ROCKY', '19768 N. ADRIAN AVE.', 'GILBERT', 'AZ', '85142', '4805557865',
'ROCKY@PALERMO.COM');

```

## Menu Table

```

CREATE TABLE MENU (
MENU_ITEM_ID
RESTOCK_LEVEL
ITEM_DESC
SERVING_SIZE
QOH
QOO
PRIMARY KEY (MENU_ITEM_ID));

```

INSERT INTO MENU
------------------

VALUES('800', '700', 'POPCORN', 'SMALL', '689', '0');
---

INSERT INTO MENU
------------------

VALUES('801', '700', 'POPCORN', 'MED', '689', '0');
---

INSERT INTO MENU
------------------

VALUES('802', '700', 'POPCORN', 'LARG', '689', '0');
--

INSERT INTO MENU
------------------

VALUES('803', '700', 'POPCORN', 'XL', '689', '0');
--

INSERT INTO MENU
------------------

VALUES('900', '500', 'SODA', 'SMALL', '412', '0');
--

INSERT INTO MENU
------------------

VALUES('901', '500', 'SODA', 'MED', '501', '0');
--

INSERT INTO MENU
------------------

VALUES('903', '500', 'SODA', 'LARG', '920', '0');
---

INSERT INTO MENU
------------------

VALUES('904', '500', 'SODA', 'XL', '880', '0');
---

INSERT INTO MENU
------------------

VALUES('111111', NULL, 'MOVIE TICKET', 'KIDS', '10000', '0');
---

INSERT INTO MENU
------------------

VALUES('222222', NULL, 'MOVIE TICKET', 'ADULT', '100000', '0');
---

INSERT INTO MENU
------------------

VALUES('333333', NULL, 'MOVIE TICKET', 'SENIOR', '100000', '0');
--

INSERT INTO MENU
------------------

VALUES('444444', NULL, 'MOVIE TICKET', 'MATINEE', '100000', '0');
---

INSERT INTO MENU
------------------

VALUES('700', '100', 'CHEESE PIZZA', '8 IN', '60', '150');
--

INSERT INTO MENU
------------------

VALUES('701', '50', 'CHEESE PIZZA', '10 IN', '160', '0');
---

INSERT INTO MENU
------------------

VALUES('702', '100', 'CHEESE PIZZA', '14 IN', '22', '200');
---

INSERT INTO MENU
------------------

VALUES('710', '100', 'VEGGIE PIZZA', '8 IN', '6', '150');
---

INSERT INTO MENU
------------------

VALUES('711', '50', 'VEGGIE PIZZA', '10 IN', '60', '0');
INSERT INTO MENU
VALUES('712', '100', 'VEGGIE PIZZA', '14 IN', '0', '200');
INSERT INTO MENU
VALUES('720', '100', 'PEPPERONI PIZZA', '8 IN', '6', '250');
INSERT INTO MENU
VALUES('721', '50', 'PEPPERONI PIZZA', '10 IN', '0', '100');
INSERT INTO MENU
VALUES('722', '100', 'PEPPERONI PIZZA', '14 IN', '120', '0');
INSERT INTO MENU
VALUES('730', '100', 'MEAT LOVERS PIZZA', '8 IN', '101', '0');
INSERT INTO MENU
VALUES('731', '50', 'MEAT LOVERS PIZZA', '10 IN', '0', '0');
INSERT INTO MENU
VALUES('732', '100', 'MEAT LOVERS PIZZA', '14 IN', '120', '0');
INSERT INTO MENU
VALUES('600', '500', 'M&M'S', NULL, '44', '0');
INSERT INTO MENU
VALUES('610', '300', 'SKITTLES', NULL, '461', '0');
INSERT INTO MENU
VALUES('620', '300', 'REDVINES', NULL, '0', '0');
INSERT INTO MENU
VALUES('601', '500', 'PEANUT M&M'S', NULL, '233', '100');
INSERT INTO MENU
VALUES('630', '200', 'JUNIOR MINTS', NULL, '393', '1000');
INSERT INTO MENU
VALUES('640', '100', 'GOOD & PLENTY', NULL, '44', '100');
INSERT INTO MENU
VALUES('550', '300', 'SOFT PRETZEL BITES W/ CHEESE', NULL, '461', '0');
INSERT INTO MENU
VALUES('551', '300', 'SOFT PREZTEL BITES W/ MUSTARD', NULL, '400', '0');
INSERT INTO MENU
VALUES('552', '50', 'SIDE OF CHEESE', NULL, '23', '0');
INSERT INTO MENU
VALUES('553', '50', 'SIDE OF MUSTARD', NULL, '13', '100');

INSERT INTO MENU
VALUES('530', '300', 'VEGGIE STICKS W/ DRESSING', NULL, '460', '0');
INSERT INTO MENU
VALUES('531', '300', 'VEGGIE STICKS W/ HUMMUS', NULL, '0', '0');
INSERT INTO MENU
VALUES('532', '50', 'SIDE OF DRESSING', NULL, '43', '0');
INSERT INTO MENU
VALUES('533', '50', 'SIDE OF HUMMUS', NULL, '33', '100');
INSERT INTO MENU
VALUES('500', '800', 'BEER', '12 OZ', '913', '500');
INSERT INTO MENU
VALUES('510', '800', 'WINE', '5 OZ', '790', '0');
INSERT INTO MENU
VALUES('520', '800', 'COCKTAIL', '1.5 OZ', '724', '0');

## Ticket Table

CREATE TABLE TICKET(			
TICKET_ID	INT(6)	NOT NULL,	
DATE_TIME_ID	INT(14)	NOT NULL,	
MOVIE_ID	INT(6)	NOT NULL,	
EMPLOYED_ID	INT(6)	NOT NULL,	
CHECK_ID	INT(6)	NOT NULL,	
PRIMARY KEY(TICKET_ID)			
FOREIGN KEY (DATE_TIME_ID) REFERENCES DATI (DATE_TIME_ID),			
FOREIGN KEY (MOVIE_ID) REFERENCES MOVIE (MOVIE_ID),			
FOREIGN KEY (EMPLOYED_ID) REFERENCES EMPLOYED (EMPLOYED_ID));			

INSERT INTO TICKET
VALUES('100001', '20121019101403', 'A10001', '000001', '200001');
INSERT INTO TICKET
VALUES('100002', '20121119081405', 'A10002', '000002', '200002');

```
INSERT INTO TICKET
VALUES('100003', '20131225073055', 'A10003','000003', '200003');

INSERT INTO TICKET
VALUES('100004', '20110825034511', 'A10004', '000004', '200004');

INSERT INTO TICKET
VALUES('100005', '20120405104509', 'A10005', '000005', '200005');

INSERT INTO TICKET
VALUES('100006', '20110304050607', 'A10006', '000006', '200006');

INSERT INTO TICKET
VALUES('100007', '20110405021055', 'A10007', '000007', '200007');

INSERT INTO TICKET
VALUES('100008', '20110607115959', 'A10008', '000008', '200008');

INSERT INTO TICKET
VALUES('100009', '20120103115425', 'A10009', '000009', '200009');

INSERT INTO TICKET
VALUES('100010', '20110606125505', 'A10010', '000010', '200010');

INSERT INTO TICKET
VALUES('100011', '20120202020202', 'A10011', '000011', '200011');

INSERT INTO TICKET
VALUES('100012', '20110405040406', 'A10012', '000012', '200012');

INSERT INTO TICKET
VALUES('100013', '20120506073434', 'A10013', '000013', '200013');

INSERT INTO TICKET
VALUES('100014', '20130101010101', 'A10014', '000014', '200014');

INSERT INTO TICKET
VALUES('100015', '20121010101010', 'A10015', '000015', '200015');

INSERT INTO TICKET
VALUES('100016', '20121203124503', 'A10016', '000016', '200016');

INSERT INTO TICKET
VALUES('100017', '20111223020202', 'A10017', '000017', '200017');

INSERT INTO TICKET
VALUES('100018', '20120404113223', 'A10018', '000018', '200018');

INSERT INTO TICKET
VALUES('100019', '20110908034554', 'A10019', '000019', '200019');
```

```
VALUES('100020', '20120807090859', 'A10020', '000020', '200020');  
INSERT INTO TICKET  
VALUES('100021', '20110606034502', 'A10021', '000021', '200021');  
INSERT INTO TICKET  
VALUES('100022', '20120909045454', 'A10022', '000022', '200022');  
INSERT INTO TICKET  
VALUES('100023', '20120402010159', 'A10023', '000023', '200023');  
INSERT INTO TICKET  
VALUES('100024', '20121212121212', 'A10024', '000024', '200024');  
INSERT INTO TICKET  
VALUES('100025', '20111111111111', 'A10025', '000025', '200025');  
INSERT INTO TICKET  
VALUES('100026', '20120111011111', 'A10026', '000026', '200026');  
INSERT INTO TICKET  
VALUES('100027', '20121230121211', 'A10027', '000027', '200027');  
INSERT INTO TICKET  
VALUES('100028', '20110101121111', 'A10028', '000028', '200028');  
INSERT INTO TICKET  
VALUES('100029', '20110304040454', 'A10029', '000029', '200029');  
INSERT INTO TICKET  
VALUES('100030', '20120405010220', 'A10030', '000030', '200030');  
INSERT INTO TICKET  
VALUES('100031', '20130101032121', 'A10031', '000031', '200031');  
INSERT INTO TICKET  
VALUES('100032', '20120807115904', 'A10032', '000032', '200032');  
INSERT INTO TICKET  
VALUES('100033', '20110505121234', 'A10033', '000033', '200033');  
INSERT INTO TICKET  
VALUES('100034', '20120405122424', 'A10034', '000034', '200034');  
INSERT INTO TICKET  
VALUES('100035', '20111231090959', 'A10035', '000035', '200035');  
INSERT INTO TICKET  
VALUES('100036', '20120203044554', 'A10036', '000036', '200036');  
INSERT INTO TICKET  
VALUES('100037', '20110405090908', 'A10037', '000037', '200037');
```

```

INSERT INTO TICKET
VALUES('100038', '20110504040550', 'A10038', '000038', '200038');

INSERT INTO TICKET
VALUES('100039', '20120810095454', 'A10039', '000039', '200039');

INSERT INTO TICKET
VALUES('100040', '20110505125505', 'A10040', '000040', '200040');

```

## Dati Table

Date and Time
CREATE TABLE DATI(
DATE_TIME_ID INT(6) NOTNULL,
SCREEN_ID INT(6) NOTNULL,
MOVIE_ID INT(6) NOTNULL,
PRIMARY KEY (SCREEN_ID, MOVIE_ID));

INSERT INTO DATI
VALUES ('20121019101403', '101', 'A10001');
INSERT INTO DATI
VALUES ('20121119081405', '102', 'A10002');
INSERT INTO DATI
VALUES ('20131225073055', '103', 'A10003');
INSERT INTO DATI
VALUES ('20110825034511', '104', 'A10004');
INSERT INTO DATI
VALUES ('20120405104509', '105', 'A10005');
INSERT INTO DATI
VALUES ('20110304050607', '106', 'A10006');
INSERT INTO DATI
VALUES ('20110405021055', '107', 'A10007');
INSERT INTO DATI
VALUES ('20110607115959', '108', 'A10008');

INSERT INTO DATI
VALUES ('20120103115425', '109', 'A10009');
INSERT INTO DATI
VALUES ('20110606125505', '110', 'A10010');
INSERT INTO DATI
VALUES ('20120202020202', '111', 'A10011');
INSERT INTO DATI
VALUES ('20110405040406', '112', 'A10012');
INSERT INTO DATI
VALUES ('20120506073434', '113', 'A10013');
INSERT INTO DATI
VALUES ('20130101010101', '114', 'A10014');
INSERT INTO DATI
VALUES ('20121010101010', '115', 'A10015');
INSERT INTO DATI
VALUES ('20121203124503', '116', 'A10016');
INSERT INTO DATI
VALUES ('20111223020202', '117', 'A10017');
INSERT INTO DATI
VALUES ('20120404113223', '118', 'A10018');
INSERT INTO DATI
VALUES ('20110908034554', '119', 'A10019');
INSERT INTO DATI
VALUES ('20120807090859', '120', 'A10020')
INSERT INTO DATI
VALUES ('20120807090859', '121', 'A10021');
INSERT INTO DATI
VALUES ('20120909045454', '122', 'A10022');
INSERT INTO DATI
VALUES ('20120402010159', '123', 'A10023');
INSERT INTO DATI
VALUES ('20121212121212', '124', 'A10024');
INSERT INTO DATI
VALUES ('20111111111111', '125', 'A10025');
INSERT INTO DATI

VALUES ('20120111011111', '126', 'A10026');
INSERT INTO DATI
VALUES ('20121230121211', '127', 'A10027');
INSERT INTO DATI
VALUES ('20110101121111', '128', 'A10028');
INSERT INTO DATI
VALUES ('20110304040454', '129', 'A10029');
INSERT INTO DATI
VALUES ('20120405010220', '130', 'A10030');
INSERT INTO DATI
VALUES ('20130101032121', '131', 'A10031');
INSERT INTO DATI
VALUES ('20120807115904', '132', 'A10032');
INSERT INTO DATI
VALUES ('20110505121234', '133', 'A10033');
INSERT INTO DATI
VALUES ('20120405122424', '134', 'A10034');
INSERT INTO DATI
VALUES ('20111231090959', '135', 'A10035');
INSERT INTO DATI
VALUES ('20120203044554', '136', 'A10036');
INSERT INTO DATI
VALUES ('20110405090908', '137', 'A10037');
INSERT INTO DATI
VALUES ('20110504040550', '138', 'A10038');
INSERT INTO DATI
VALUES ('20120810095454', '139', 'A10039');
INSERT INTO DATI
VALUES ('20110505125505', '140', 'A10040');

## Employed Table

CREATE TABLE
--------------

EMPLOYED (		
EMPLOYED_ID	CHAR(6)	NOT NULL,
F_NAME	VARCHAR(35)	NOT NULL,
INITIAL	CHAR(1)	NOT NULL,
L_NAME	VARCHAR(35)	NOT NULL,
ADDRESS	VARCHAR(35)	NOT NULL,
CITY	VARCHAR(35)	NOT NULL,
STATE	CHAR(2)	NOT NULL,
ZIP	CHAR(5)	NOT NULL,
PHN_NUM	CHAR(10)	NOT NULL,
EMAIL	VARCHAR(35)	NOT NULL);
MANAGER_ID		
THEATRE_ID		
PRIMARY KEY (EMPLOYED_ID)		

INSERT INTO EMPLOYED
VALUES('000001', 'SARA', 'R', 'JONES', '1234 W MAIN ST.', 'TEMPE', 'AZ', '85001', '6023457890', 'SARAH.JONES@GMAIL.COM', '10', '112');
INSERT INTO EMPLOYED
VALUES('000002', 'JANE', 'P', 'LEE', '1231 S GEORGIA ST.', 'TEMPE', 'AZ', '85001', '4807678544', 'JANE.LEE@GMAIL.COM', '9', '111');
INSERT INTO EMPLOYED
VALUES('000003', 'ANNE', 'E', 'PRIOR', '2405 E WARNER ST.', 'TEMPE', 'AZ', '85003', '4807678574', 'ANNE.PRIOR@GMAIL.COM', '8', '110');
INSERT INTO EMPLOYED
VALUES('000004', 'THOMAS', 'M', 'SMITH', '2468 S LOVEBIRD ST.', 'TEMPE', 'AZ', '85005', '4808956677', 'THOMAS.SMITH@GMAIL.COM', '7', '109');
INSERT INTO EMPLOYED
VALUES('000005', 'JACOB', 'S', 'MILLER', '2440 E 64TH ST.', 'TEMPE', 'AZ', '85005', '6027676544', 'JACOB.MILLER@GMAIL.COM', '6', '108');
INSERT INTO EMPLOYED
VALUES('000006', 'ELI', 'D', 'JOHNSON', '1807 W PEPPER ST.', 'TEMPE', 'AZ', '85005', '4809753356', 'ELI.JOHNSON@GMAIL.COM', '5', '107');

```
INSERT INTO EMPLOYED
VALUES('000007', 'SANTIAGO', 'B', 'REES', '1446 E GUADALUPE ST.', 'TEMPE', 'AZ', '85003',
'6024102347', 'SANTIAGO.REES@GMAIL.COM','4','106');

INSERT INTO EMPLOYED
VALUES('000008', 'LILLY', 'H', 'COOK', '1447 E TERESA ST.', 'TEMPE', 'AZ', '85004', '6028534599',
'LILLY.COOK@GMAIL.COM','3','105');

INSERT INTO EMPLOYED
VALUES('000009', 'ANDREW', 'W', 'JAMES', '1506 S TEMPE ST.', 'TEMPE', 'AZ', '85004', '6025679034',
'ANDREW.JAMES@GMAIL.COM','2','104');

INSERT INTO EMPLOYED
VALUES('000010', 'PHOEBE', 'R', 'MOSS', '2891 E CARMEL ST.', 'TEMPE', 'AZ', '85005', '4806785644',
'PHOEBE.MOSS@GMAIL.COM','1','103');

INSERT INTO EMPLOYED
VALUES('000011', 'PIPER', 'L', 'LUND', '1698 W GOLD ST.', 'TEMPE', 'AZ', '85001', '6026782091',
'PIPER.LUND@GMAIL.COM','2','102');

INSERT INTO EMPLOYED
VALUES('000012', 'PENELOPE', 'O', 'IRELAND', '1698 W GOLD ST.', 'TEMPE', 'AZ', '85003',
'6029007500', 'PENELOPE.IRELAND@GMAIL.COM','3','101');

INSERT INTO EMPLOYED
VALUES('000013', 'JOHN', 'A', 'NEWT', '1534 E 34TH ST.', 'TEMPE', 'AZ', '85001', '6029980533',
'JOHN.NEWT@GMAIL.COM','4','100');

INSERT INTO EMPLOYED
VALUES('000014', 'ELLE', 'J', 'MOOSE', '1700 W STAR ST.', 'TEMPE', 'AZ', '85002', '6020035411',
'ELLE.MOOSE@GMAIL.COM','5','101');

INSERT INTO EMPLOYED
VALUES('000015', 'MICHAEL', 'L', 'SCARN', '2788 E POOL ST.', 'TEMPE', 'AZ', '85005', '4807564429',
'MICHAEL.SCARN@GMAIL.COM','6','102');

INSERT INTO EMPLOYED
VALUES('000016', 'SKYE', 'O', 'CARLSON', '1400 E OPAL ST.', 'TEMPE', 'AZ', '85002', '6020035411',
'SKYE.CARLSON@GMAIL.COM','7','103');

INSERT INTO EMPLOYED
VALUES('000017', 'ARCHER', 'E', 'MONSON', '2806 E LINDRICK ST.', 'TEMPE', 'AZ', '85004',
'6029085564', 'ARCHER.MONSON@GMAIL.COM','8','104');

INSERT INTO EMPLOYED
VALUES('000018', 'LANE', 'B', 'DESPAIN', '1399 W JUSTICE ST.', 'TEMPE', 'AZ', '85004', '6027869933',
'LANE.DESPAN@GMAIL.COM','9','105');

INSERT INTO EMPLOYED
VALUES('000019', 'UMA', 'T', 'STRONG', '1569 W STRAW ST.', 'TEMPE', 'AZ', '85004', '6029086112',
```

```
'UMA.STRONG@GMAIL.COM','10','106');

INSERT INTO EMPLOYED

VALUES('000020', 'BILLY', 'W', 'MCDONALD', '1776 E INDEPENDENCE ST.', 'TEMPE', 'AZ', '85005',
'6027089376', 'BILLY.MCDONALD@GMAIL.COM','11','107');

INSERT INTO EMPLOYED

VALUES('000021', 'MANNY', 'P', 'BLUE', '2011 E YOUNG ST.', 'TEMPE', 'AZ', '85002', '6028906134',
'MANNY.BLUE@GMAIL.COM','12','108');

INSERT INTO EMPLOYED

VALUES('000022', 'ERIN', 'L', 'DESTINY', '1592 S TRUE ST.', 'TEMPE', 'AZ', '85002', '6023851290',
'ERIN.DESTINY@GMAIL.COM','11','109');

INSERT INTO EMPLOYED

VALUES('000023', 'POLLY', 'B', 'MCARTHUR', '1224 E STRAWBERRY ST.', 'TEMPE', 'AZ', '85001',
'4807162253', 'POLLY.MCARTHUR@GMAIL.COM','10','110');

INSERT INTO EMPLOYED

VALUES('000024', 'BRANDON', 'J', 'BUNCH', '3907 W TREE ST.', 'TEMPE', 'AZ', '85005', '6026904321',
'BRANDON.BUNCH@GMAIL.COM','9','111');

INSERT INTO EMPLOYED

VALUES('000025', 'KATYA', 'S', 'BERRY', '3400 S APPLETREE ST.', 'TEMPE', 'AZ', '85004',
'6029011542', 'KATYA.BERRY@GMAIL.COM','8','112');

INSERT INTO EMPLOYED

VALUES('000026', 'STEVEN', 'C', 'AUSTIN', '2300 N HOUSTON ST.', 'TEMPE', 'AZ', '85004',
'4806709327', 'STEVEN.AUSTIN@GMAIL.COM','7','111');

INSERT INTO EMPLOYED

VALUES('000027', 'KATIE', 'R', 'CHRISTIANSEN', '2809 E COOKIE ST.', 'TEMPE', 'AZ', '85002',
'6020997184', 'KATIE.CHRISTIANSEN@GMAIL.COM','6','110');

INSERT INTO EMPLOYED

VALUES('000028', 'NINA', 'W', 'BALLARD', '4003 S LIME ST.', 'TEMPE', 'AZ', '85001', '6020021185',
'NINA.BALLARD@GMAIL.COM','5','109');

INSERT INTO EMPLOYED

VALUES('000029', 'GREG', 'F', 'RIGGS', '1875 E RURAL ST.', 'TEMPE', 'AZ', '85006', '6028471904',
'GREG.RIGGS@GMAIL.COM','4','108');

INSERT INTO EMPLOYED

VALUES('000030', 'CRISTY', 'A', 'VICTOR', '4560 E SILVER ST.', 'TEMPE', 'AZ', '85002', '4807264002',
'CHRISTY.VICTOR@GMAIL.COM','3','107');

INSERT INTO EMPLOYED

VALUES('000031', 'STELLA', 'O', 'RICKS', '3608 W YUMA ST.', 'TEMPE', 'AZ', '85002', '4800096383',
'STELLA.RICKS@GMAIL.COM','2','106');

INSERT INTO EMPLOYED
```

```

VALUES('000032', 'CARRIE', 'P', 'LION', '3900 E BROADWAY ST.', 'TEMPE', 'AZ', '85005', '4809087766',
'CARRIE.LION@GMAIL.COM','1','105');

INSERT INTO EMPLOYED

VALUES('000033', 'BLAKE', 'R', 'GARRETT', '900 E SYDNEY ST.', 'TEMPE', 'AZ', '85004', '480948372',
'BLAKE.GARETT@GMAIL.COM','2','104');

INSERT INTO EMPLOYED

VALUES('000034', 'ANNIE', 'E', 'OWENS', '4700 W MOON ST.', 'TEMPE', 'AZ', '85002', '6023459086',
'ANNIE.OWENS@GMAIL.COM','3','103');

INSERT INTO EMPLOYED

VALUES('000035', 'SUNNY', 'K', 'SPENCER', '2740 E PILOT ST.', 'TEMPE', 'AZ', '85005', '4809382494',
'SUNNY.SPENCER@GMAIL.COM','4','102');

INSERT INTO EMPLOYED

VALUES('000036', 'FRED', 'G', 'BUBBLES', '2668 W TEA ST.', 'TEMPE', 'AZ', '85003', '4808045829',
'FRED.BUBBLES@GMAIL.COM','5','101');

INSERT INTO EMPLOYED

VALUES('000037', 'HALEY', 'E', 'SANDERS', '4508 S ROSE ST.', 'TEMPE', 'AZ', '85005', '4808903348',
'HALEY.SANDERS@GMAIL.COM','6','100');

INSERT INTO EMPLOYED

VALUES('000038', 'HARRY', 'R', 'GRAHAM', '100 S IGLOO ST.', 'TEMPE', 'AZ', '85005', '4808007182',
'HARRY.GRAHAM@GMAIL.COM','7','101');

INSERT INTO EMPLOYED

VALUES('000039', 'CHARLIE', 'C', 'WALKER', '7200 S BULLHORN ST.', 'TEMPE', 'AZ', '85005',
'4801110972', 'CHARLIE.WALKER@GMAIL.COM','8','102');

INSERT INTO EMPLOYED

VALUES('000040', 'MIA', 'B', 'FOX', '3705 E CALICO ST.', 'TEMPE', 'AZ', '85005', '4800804902',
'MIA.FOX@GMAIL.COM','9','103');

```

## Screen Table

Screen
CREATE TABLE SCREEN(
SCREEN_ID INT(6) NOTNULL,
PRIMARY KEY (SCREEN_ID));

INSERT INTO SCREEN VALUES('101')
----------------------------------

INSERT INTO SCREEN VALUES('102')
INSERT INTO SCREEN VALUES('103')
INSERT INTO SCREEN VALUES('104')
INSERT INTO SCREEN VALUES('105')
INSERT INTO SCREEN VALUES('106')
INSERT INTO SCREEN VALUES('107')
INSERT INTO SCREEN VALUES('108')
INSERT INTO SCREEN VALUES('109')
INSERT INTO SCREEN VALUES('110')
INSERT INTO SCREEN VALUES('111')
INSERT INTO SCREEN VALUES('112')
INSERT INTO SCREEN VALUES('113')
INSERT INTO SCREEN VALUES('114')
INSERT INTO SCREEN VALUES('115')
INSERT INTO SCREEN VALUES('116')
INSERT INTO SCREEN VALUES('117')
INSERT INTO SCREEN VALUES('118')
INSERT INTO SCREEN VALUES('119')
INSERT INTO SCREEN VALUES('120')
INSERT INTO SCREEN VALUES('121')
INSERT INTO SCREEN VALUES('122')
INSERT INTO SCREEN VALUES('123')
INSERT INTO SCREEN VALUES('124')
INSERT INTO SCREEN VALUES('125')
INSERT INTO SCREEN VALUES('126')
INSERT INTO SCREEN VALUES('127')
INSERT INTO SCREEN VALUES('128')
INSERT INTO SCREEN VALUES('129')
INSERT INTO SCREEN VALUES('130')
INSERT INTO SCREEN VALUES('131')
INSERT INTO SCREEN VALUES('132')
INSERT INTO SCREEN VALUES('133')
INSERT INTO SCREEN VALUES('134')
INSERT INTO SCREEN VALUES('135')
INSERT INTO SCREEN VALUES('136')

INSERT INTO SCREEN VALUES('137')
INSERT INTO SCREEN VALUES('138')
INSERT INTO SCREEN VALUES('139')
INSERT INTO SCREEN VALUES('140')

## Movie Table

Movie
CREATE TABLE MOVIE(
MOVIE_ID INT(6),
TITLE VARCHAR(35) NOTNULL,
RATING VARCHAR(5) NOTNULL,
RELEASE_YR DATE NOTNULL,
PRIMARY KEY (MOVIE_ID));

INSERT INTO MOVIE VALUES ('A10001', 'HARRY POTTER & THE SORCERS STONE', 'PG', '2001');
INSERT INTO MOVIE VALUES ('A10002', 'TITANIC', 'PG-13', '1997');
INSERT INTO MOVIE VALUES ('A10003', 'STAR WARS EPISODE IX', 'PG-13', '2019');
INSERT INTO MOVIE VALUES ('A10004', 'BACK TO THE FUTURE', 'PG', '1985');
INSERT INTO MOVIE VALUES ('A10005', 'JAWS', 'PG', '1975');
INSERT INTO MOVIE VALUES ('A10006', 'JURASSIC PARK', 'PG-13', '1993');
INSERT INTO MOVIE VALUES ('A10007', 'GHOST', 'PG-13', '1990');
INSERT INTO MOVIE VALUES ('A10008', 'THE LION KING', 'G', '1994');
INSERT INTO MOVIE VALUES ('A10009', 'TOY STORY', 'G', '1995');
INSERT INTO MOVIE VALUES ('A10010', 'THE AVENGERS ', 'PG-13', '2012');
INSERT INTO MOVIE VALUES ('A10011', 'GREASE', 'PG', '1978');
INSERT INTO MOVIE VALUES ('A10012', 'ROCKY', 'PG', '1976');
INSERT INTO MOVIE VALUES ('A10013', 'HARRY POTTER & CHAMBER OF SECRETS', 'PG', '2002');
INSERT INTO MOVIE VALUES ('A10014', 'HARRY POTTER & PRISONER OF AZKABAN', 'PG', '2004');
INSERT INTO MOVIE VALUES ('A10015', 'HARRY POTTER & GOBLET OF FIRE', 'PG-13', '2005');
INSERT INTO MOVIE VALUES ('A10016', 'HARRY POTTER & ORDER OF THE PHOENIX', 'PG-13', '2007');
INSERT INTO MOVIE VALUES ('A10017', 'HARRY POTTER & HALF BLOOD PRINCE', 'PG', '2009');

```

INSERT INTO MOVIE VALUES ('A10018', 'HARRY POTTER & DEATHLY HALLOWS P1', 'PG-13', '2010');
INSERT INTO MOVIE VALUES ('A10019', 'HARRY POTTER & DEATHLY HALLOWS P2', 'PG-13', '2011');
INSERT INTO MOVIE VALUES ('A10020', 'STAR WARS', 'PG', '1977');
INSERT INTO MOVIE VALUES ('A10021', 'STAR WARS EPISODE V', 'PG', '1980');
INSERT INTO MOVIE VALUES ('A10022', 'STAR WARS EPISODE VI', 'PG-13', '1983');
INSERT INTO MOVIE VALUES ('A10023', 'STAR WARS EPISODE I', 'PG', '1999');
INSERT INTO MOVIE VALUES ('A10024', 'STAR WARS EPISODE II', 'PG', '2002');
INSERT INTO MOVIE VALUES ('A10025', 'STAR WARS EPISODE III', 'PG', '2005');
INSERT INTO MOVIE VALUES ('A10026', 'STAR WARS EPISODE VII', 'PG-13', '2015');
INSERT INTO MOVIE VALUES ('A10027', 'STAR WARS EPISODE VIII', 'PG-13', '2017');
INSERT INTO MOVIE VALUES ('A10028', 'THE AVENGERS: AGE OF ULTRON', 'PG-13', '2015');
INSERT INTO MOVIE VALUES ('A10029', 'TOY STORY 2', 'G', '1999');
INSERT INTO MOVIE VALUES ('A10030', 'TOY STORY 3', 'G', '2010');
INSERT INTO MOVIE VALUES ('A10031', 'TOY STORY 4', 'G', '2019');
INSERT INTO MOVIE VALUES ('A10032', 'THE AVENGERS: INFINITY WAR', 'PG-13', '2018');
INSERT INTO MOVIE VALUES ('A10033', 'THE AVENGERS: END GAME', 'PG-13', '2019');
INSERT INTO MOVIE VALUES ('A10034', 'BACK TO THE FUTURE P2', 'PG', '1989');
INSERT INTO MOVIE VALUES ('A10035', 'BACK TO THE FUTURE P3', 'PG', '1990');
INSERT INTO MOVIE VALUES ('A10036', 'KILL BILL: VOL 1', 'R', '2003');
INSERT INTO MOVIE VALUES ('A10036', 'KILL BILL: VOL 2', 'R', '2004');
INSERT INTO MOVIE VALUES ('A10038', 'INGLORIOUS BASTERDS', 'R', '2009');
INSERT INTO MOVIE VALUES ('A10039', 'SLEEPING BEAUTY', 'G', '1959');
INSERT INTO MOVIE VALUES ('A10040', 'SNOW WHITE & THE SEVEN DWARVES', 'G', '1937');

```

## Invoice Table

CREATE TABLE INVOICE(		
CHECK_ID	CHAR(6)	PRIMARY KEY
CHECK_TYPE	VARCHAR(35)	NOT NULL,
TOTAL	DECIMAL (3,2)	NOT NULL,

DISCOUNT	DECIMAL (3,2)	NOT NULL,
TAX	DECIMAL (3,2)	NOT NULL,
TIP	DECIMAL (3,2)	NOT NULL,
<i>EMPLOYED_ID</i>	INT(6)	NOT NULL,
<i>DATE_TIME_ID</i>	INT(6)	NOT NULL,
<i>ORDER_ID</i>	INT(6)	NOT NULL,
EMPLOYED_ID CHAR(6) FOREIGN KEY REFERENCES EMPLOYED (EMPLOYED_ID),		
DATE_TIME_IS CHAR(6) FOREIGN KEY REFERENCES DATE AND TIME (DATE_TIME_ID),		
ORDER_ID CHAR(6) FOREIGN KEY REFERENCES ORDERS (ORDER_ID);		

INSERT INTO INVOICE
VALUES('200001', 'SNACK STAND', '12.00', '0.00', '0.72', '2.40', '000001', '20110101121111', '600001');
INSERT INTO INVOICE
VALUES('200002', 'SNACK STAND', '7.00', '0.00', '0.42', '1.40', '000002', '20110304040454', '600002');
INSERT INTO INVOICE
VALUES('200003', 'SNACK STAND', '8.00', '0.00', '0.48', '1.60', '000003', '20110304040454', '600003');
INSERT INTO INVOICE
VALUES('200004', 'SNACK STAND', '27.00', '2.70', '1.62', '5.40', '000004', '20110405021055', '600004');
INSERT INTO INVOICE
VALUES('200005', 'SNACK STAND', '5.50', '0.00', '0.33', '1.1', '000005', '20110405040406', '600005');
INSERT INTO INVOICE
VALUES('200006', 'SNACK STAND', '6.00', '0.00', '0.36', '1.20', '000006', '20110405090908', '600006');
INSERT INTO INVOICE
VALUES('200007', 'SNACK STAND', '13.00', '0.00', '0.78', '2.60', '000007', '20110504040550', '600007');
INSERT INTO INVOICE
VALUES('200008', 'SNACK STAND', '7.00', '0.00', '0.42', '1.40', '000008', '20110505121234', '600008');

INSERT INTO INVOICE
VALUES('200009', 'SNACK STAND', '15.00', '1.50', '0.90', '3.00', '000009', '20110505125505', '600009');
INSERT INTO INVOICE
VALUES('200010', 'TICKET COUNTER', '11.00', '0.00', '0.66', '2.20', '000010', '20110606034502', '600010');
INSERT INTO INVOICE
VALUES('200011', 'TICKET COUNTER', '28.00', '2.80', '1.68', '5.60', '000011', '20110606125505', '600011');
INSERT INTO INVOICE
VALUES('200012', 'TICKET COUNTER', '12.00', '0.00', '0.72', '2.40', '000012', '20110607115959', '600012');
INSERT INTO INVOICE
VALUES('200013', 'TICKET COUNTER', '39.00', '3.90', '2.34', '7.8', '000013', '20110825034511', '600013');
INSERT INTO INVOICE
VALUES('200014', 'SNACK STAND', '7.00', '0.00', '0.42', '1.40', '000014', '20110908034554', '600014');
INSERT INTO INVOICE
VALUES('200015', 'SNACK STAND', '7.50', '0.00', '0.34', '1.20', '000015', '20111111111111', '600015');
INSERT INTO INVOICE
VALUES('200016', 'SNACK STAND', '16.00', '1.60', '0.96', '3.20', '000016', '2011223020202', '600016');
INSERT INTO INVOICE
VALUES('200017', 'SNACK STAND', '7.00', '0.00', '0.42', '1.40', '000017', '2011231090959', '600017');
INSERT INTO INVOICE
VALUES('200018', 'SNACK STAND', '7.50', '0.00', '0.34', '1.20', '000018', '20120103115425', '600018');
INSERT INTO INVOICE
VALUES('200019', 'SNACK STAND', '8.00', '0.00', '0.48', '1.60', '000019', '20120202020202', '600019');
INSERT INTO INVOICE
VALUES('200020', 'SNACK STAND', '21.00', '2.10', '1.26', '4.20', '000020', '20120203044554', '600020');
INSERT INTO INVOICE
VALUES('200021', 'SNACK STAND', '30.00', '3.00', '1.80', '6.00', '000021', '20120402010159', '600021');
INSERT INTO INVOICE
VALUES('200022', 'SNACK STAND', '8.00', '0.00', '0.48', '1.60', '000022', '20120404113223', '600022');
INSERT INTO INVOICE
VALUES('200023', 'SNACK STAND', '7.00', '0.00', '0.42', '1.40', '000023', '20120405010220', '600023');
INSERT INTO INVOICE
VALUES('200024', 'SNACK STAND', '30.00', '3.00', '1.80', '6.00', '000024', '20120405104509', '600024');

INSERT INTO INVOICE
VALUES('200025', 'SNACK STAND', '8.00', '0.00', '0.48', '1.60', '000025', '20120405122424', '600025');
INSERT INTO INVOICE
VALUES('200026', 'SNACK STAND', '12.00', '0.00', '0.72', '2.40', '000026', '20120506073434', '600026');
INSERT INTO INVOICE
VALUES('200027', 'SNACK STAND', '4.0', '0.00', '0.24', '0.80', '000027', '20120807090859', '600027');
INSERT INTO INVOICE
VALUES('200028', 'SNACK STAND', '8.00', '0.00', '0.48', '1.60', '000028', '20120807115904', '600028');
INSERT INTO INVOICE
VALUES('200029', 'SNACK STAND', '4.00', '0.00', '0.24', '0.80', '000029', '20120810095454', '600029');
INSERT INTO INVOICE
VALUES('200030', 'SNACK STAND', '4.00', '0.00', '0.24', '0.80', '000030', '20120909045454', '600030');
INSERT INTO INVOICE
VALUES('200031', 'SNACK STAND', '20.00', '2.00', '1.20', '4.00', '000031', '20121010101010', '600031');
INSERT INTO INVOICE
VALUES('200032', 'SNACK STAND', '5.00', '0.00', '0.30', '1.00', '000032', '20121019101403', '600032');
INSERT INTO INVOICE
VALUES('200033', 'SNACK STAND', '15.00', '1.50', '0.90', '3.00', '000033', '20121119081405', '600033');
INSERT INTO INVOICE
VALUES('200034', 'SNACK STAND', '0.50', '0.00', '0.03', '0.00', '000034', '20121203124503', '600034');
INSERT INTO INVOICE
VALUES('200035', 'SNACK STAND', '1.00', '0.00', '0.06', '0.00', '000035', '20121212121212', '600035');
INSERT INTO INVOICE
VALUES('200036', 'SNACK STAND', '1.50', '0.00', '0.06', '0.00', '000036', '20121230121211', '600036');
INSERT INTO INVOICE
VALUES('200037', 'SNACK STAND', '6.00', '0.00', '0.36', '1.20', '000037', '20130101010101', '600037');
INSERT INTO INVOICE
VALUES('200038', 'SNACK STAND', '0.50', '0.00', '0.03', '0.00', '000038', '20130101032121', '600038');
INSERT INTO INVOICE
VALUES('200039', 'SNACK STAND', '1.00', '0.00', '0.06', '0.20', '000039', '20131225073055', '600039');
INSERT INTO INVOICE
VALUES('200040', 'BAR', '2.00', '0.00', '0.12', '0.00', '000040', '20110405090908', '600040');
INSERT INTO INVOICE
VALUES('200041', 'BAR', '5.00', '0.00', '0.30', '1.00', '000031', '20120404113223', '600041');
INSERT INTO INVOICE

```
VALUES('200042', 'BAR', '10.00', '0.00', '0.60', '2.00', '000032', '20121212121212', '600042');
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100002'
```

```
WHERE CHECK_ID='200002';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100003'
```

```
WHERE CHECK_ID='200003';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100004'
```

```
WHERE CHECK_ID='200004';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100005'
```

```
WHERE CHECK_ID='200005';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100006'
```

```
WHERE CHECK_ID='200006';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100007'
```

```
WHERE CHECK_ID='200007';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100008'
```

```
WHERE CHECK_ID='200008';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100009'
```

```
WHERE CHECK_ID='200009';
```

```
UPDATE INVOICE  
SET TICKET_ID='100010'  
WHERE CHECK_ID='200010';
```

```
UPDATE INVOICE  
SET TICKET_ID='100011'  
WHERE CHECK_ID='200011';
```

```
UPDATE INVOICE  
SET TICKET_ID='100012'  
WHERE CHECK_ID='200012';
```

```
UPDATE INVOICE  
SET TICKET_ID='100013'  
WHERE CHECK_ID='200013';
```

```
UPDATE INVOICE  
SET TICKET_ID='100014'  
WHERE CHECK_ID='200014';
```

```
UPDATE INVOICE  
SET TICKET_ID='100015'  
WHERE CHECK_ID='200015';
```

```
UPDATE INVOICE  
SET TICKET_ID='100016'  
WHERE CHECK_ID='200016';
```

```
UPDATE INVOICE  
SET TICKET_ID='100017'  
WHERE CHECK_ID='200017';
```

```
UPDATE INVOICE  
SET TICKET_ID='100018'  
WHERE CHECK_ID='200018';
```

UPDATE INVOICE

SET TICKET\_ID='100019'

WHERE CHECK\_ID='200019';

UPDATE INVOICE

SET TICKET\_ID='100020'

WHERE CHECK\_ID='200020';

UPDATE INVOICE

SET TICKET\_ID='100021'

WHERE CHECK\_ID='200021';

UPDATE INVOICE

SET TICKET\_ID='100022'

WHERE CHECK\_ID='200022';

UPDATE INVOICE

SET TICKET\_ID='100023'

WHERE CHECK\_ID='200023';

UPDATE INVOICE

SET TICKET\_ID='100024'

WHERE CHECK\_ID='200024';

UPDATE INVOICE

SET TICKET\_ID='100025'

WHERE CHECK\_ID='200025';

UPDATE INVOICE

SET TICKET\_ID='100026'

WHERE CHECK\_ID='200026';

UPDATE INVOICE

SET TICKET\_ID='100027'

```
WHERE CHECK_ID='200027';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100028'
```

```
WHERE CHECK_ID='200028';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100029'
```

```
WHERE CHECK_ID='200029';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100030'
```

```
WHERE CHECK_ID='200030';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100031'
```

```
WHERE CHECK_ID='200031';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100032'
```

```
WHERE CHECK_ID='200032';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100033'
```

```
WHERE CHECK_ID='200033';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100034'
```

```
WHERE CHECK_ID='200034';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100035'
```

```
WHERE CHECK_ID='200035';
```

```
UPDATE INVOICE
```

```
SET TICKET_ID='100036'  
WHERE CHECK_ID='200036';
```

UPDATE INVOICE

```
SET TICKET_ID='100037'  
WHERE CHECK_ID='200037';
```

UPDATE INVOICE

```
SET TICKET_ID='100038'  
WHERE CHECK_ID='200038';
```

UPDATE INVOICE

```
SET TICKET_ID='100039'  
WHERE CHECK_ID='200039';
```

UPDATE INVOICE

```
SET TICKET_ID='100040'  
WHERE CHECK_ID='200040';
```

UPDATE INVOICE

```
SET TICKET_ID='100041'  
WHERE CHECK_ID='200041';
```

UPDATE INVOICE

```
SET TICKET_ID='100042'  
WHERE CHECK_ID='200042';
```

UPDATE INVOICE

```
SET DATE_TIME_ID='20121019101403'  
WHERE TICKET_ID = '100001';
```

UPDATE INVOICE

```
SET DATE_TIME_ID='20121119081405'  
WHERE TICKET_ID = '100002';
```

UPDATE INVOICE

```
SET DATE_TIME_ID='20131225073055'  
WHERE TICKET_ID = '100003;  
UPDATE INVOICE  
SET DATE_TIME_ID='20110825034511'  
WHERE TICKET_ID = '100004';  
UPDATE INVOICE  
SET DATE_TIME_ID='20120405104509'  
WHERE TICKET_ID = '100005';  
UPDATE INVOICE  
SET DATE_TIME_ID='20110304050607'  
WHERE TICKET_ID = '100006';  
UPDATE INVOICE  
SET DATE_TIME_ID='20110405021055'  
WHERE TICKET_ID = '100007';  
UPDATE INVOICE  
SET DATE_TIME_ID='20110607115959'  
WHERE TICKET_ID = '100008';  
UPDATE INVOICE  
SET DATE_TIME_ID='20120103115425'  
WHERE TICKET_ID = '100009';  
UPDATE INVOICE  
SET DATE_TIME_ID='20110606125505'  
WHERE TICKET_ID = '100010';
```

```
UPDATE INVOICE  
SET DATE_TIME_ID='20120202020202'  
WHERE TICKET_ID = '100011';  
UPDATE INVOICE  
SET DATE_TIME_ID='20110405040406'  
WHERE TICKET_ID = '100012';  
UPDATE INVOICE  
SET DATE_TIME_ID='20120506073434'  
WHERE TICKET_ID = '100013';  
UPDATE INVOICE
```

```
SET DATE_TIME_ID='20130101010101'  
WHERE TICKET_ID = '100014';  
UPDATE INVOICE  
SET DATE_TIME_ID='20121010101010'  
WHERE TICKET_ID = '100015';  
UPDATE INVOICE  
SET DATE_TIME_ID='20121203124503'  
WHERE TICKET_ID = '100016';  
UPDATE INVOICE  
SET DATE_TIME_ID='20111223020202'  
WHERE TICKET_ID = '100017';  
UPDATE INVOICE  
SET DATE_TIME_ID='20120404113223'  
WHERE TICKET_ID = '100018';  
UPDATE INVOICE  
SET DATE_TIME_ID='20110908034554'  
WHERE TICKET_ID = '100019';  
UPDATE INVOICE  
SET DATE_TIME_ID='20120807090859'  
WHERE TICKET_ID = '100020';
```

```
UPDATE INVOICE  
SET DATE_TIME_ID='20110606034502'  
WHERE TICKET_ID = '100021';  
UPDATE INVOICE  
SET DATE_TIME_ID='20120909045454'  
WHERE TICKET_ID = '100022';  
UPDATE INVOICE  
SET DATE_TIME_ID='20120402010159'  
WHERE TICKET_ID = '100023';  
UPDATE INVOICE  
SET DATE_TIME_ID='20121212121212'  
WHERE TICKET_ID = '100024';  
UPDATE INVOICE
```

```
SET DATE_TIME_ID='20111111111111'  
WHERE TICKET_ID = '100025';  
UPDATE INVOICE  
SET DATE_TIME_ID='20120111011111'  
WHERE TICKET_ID = '100026';  
UPDATE INVOICE  
SET DATE_TIME_ID='20121230121211'  
WHERE TICKET_ID = '100027';  
UPDATE INVOICE  
SET DATE_TIME_ID='20110101121111'  
WHERE TICKET_ID = '100028';  
UPDATE INVOICE  
SET DATE_TIME_ID='20110304040454'  
WHERE TICKET_ID = '100029';  
UPDATE INVOICE  
SET DATE_TIME_ID='20120405010220'  
WHERE TICKET_ID = '100030';
```

```
UPDATE INVOICE  
SET DATE_TIME_ID='20130101032121'  
WHERE TICKET_ID = '100031';  
UPDATE INVOICE  
SET DATE_TIME_ID='20120807115904'  
WHERE TICKET_ID = '100032';  
UPDATE INVOICE  
SET DATE_TIME_ID='20110505121234'  
WHERE TICKET_ID = '100033';  
UPDATE INVOICE  
SET DATE_TIME_ID='20120405122424'  
WHERE TICKET_ID = '100034';  
UPDATE INVOICE  
SET DATE_TIME_ID='20111231090959'  
WHERE TICKET_ID = '100035';  
UPDATE INVOICE
```

```

SET DATE_TIME_ID='20120203044554'
WHERE TICKET_ID = '100036';

UPDATE INVOICE

SET DATE_TIME_ID='20110405090908'
WHERE TICKET_ID = '100037';

UPDATE INVOICE

SET DATE_TIME_ID='20110504040550'
WHERE TICKET_ID = '100038';

UPDATE INVOICE

SET DATE_TIME_ID='20120810095454'
WHERE TICKET_ID = '100039';

UPDATE INVOICE

SET DATE_TIME_ID='20110505125505'
WHERE TICKET_ID = '100040';

```

<b>Orders</b>					
CREATE TABLE ORDERS(				ALTER TABLE INVOICE	
ORDER_ID	CHAR(6)	PRIMARY KEY,		ADD ORDER_ID CHAR(6) NOT NULL	
QUANTITY	INT	NOT NULL,		FOREIGN KEY REFERENCES ORDERS;	
COST	DECIMAL(6,2)	NOT NULL,			
EXT_COST	DECIMAL(6,2)	NOT NULL,			

```

MENU_ITEM_ID CHAR(6) FOREIGN KEY REFERENCES MENU (MENU_ITEM_ID),
CHECK_ID CHAR(6) FOREIGN KEY REFERENCES INVOICE (CHECK_ID));

```

```
INSERT INTO ORDERS (CHECK_ID)
```

```
VALUES (200001);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200002);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200003);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200004);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200005);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200006);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200007);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200008);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200009);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200010);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200011);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200012);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200013);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200014);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200015);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200016);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200017);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200018);
```

```
INSERT INTO ORDERS (CHECK_ID)
VALUES (200019);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200020);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200021);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200022);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200023);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200024);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200025);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200026);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200027);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200028);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200029);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200030);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200031);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200032);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200033);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200034);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200035);
INSERT INTO ORDERS (CHECK_ID)
```

```

VALUES (200036);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200037);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200038);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200039);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200040);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200041);
INSERT INTO ORDERS (CHECK_ID)
VALUES (200042);

```

```

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600001', '2', '6.00', '12.00', '800');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600002', '1', '7.00', '7.00', '801');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600003', '1', '8.00', '8.00', '802');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600004', '3', '9.00', '27.00', '803');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600005', '1', '5.50', '5.50', '900');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600006', '1', '6.00', '6.00', '901');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600007', '2', '6.50', '13.00', '902');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600008', '1', '7.00', '7.00', '903');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600009', '2', '7.50', '15.00', '904');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600010', '1', '11.00', '11.00', '111111');

```

```
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600011', '2', '14.00', '28.00', '222222');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600012', '1', '12.00', '12.00', '333333');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600013', '3', '13.00', '39.00', '444444');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600014', '1', '7.00', '7.00', '700');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600015', '1', '7.50', '7.50', '701');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600016', '2', '8.00', '16.00', '702');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600017', '1', '7.00', '7.00', '710');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600018', '1', '7.50', '7.50', '711');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600019', '1', '8.00', '8.00', '712');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600020', '3', '7.00', '21.00', '720');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600021', '4', '7.50', '30.00', '721');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600022', '1', '8.00', '8.00', '722');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600023', '1', '7.00', '7.00', '730');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600024', '4', '7.50', '30.00', '731');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600025', '1', '8.00', '8.00', '732');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600026', '3', '4.00', '12.00', '600');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600027', '1', '4.00', '4.00', '610');

INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
```

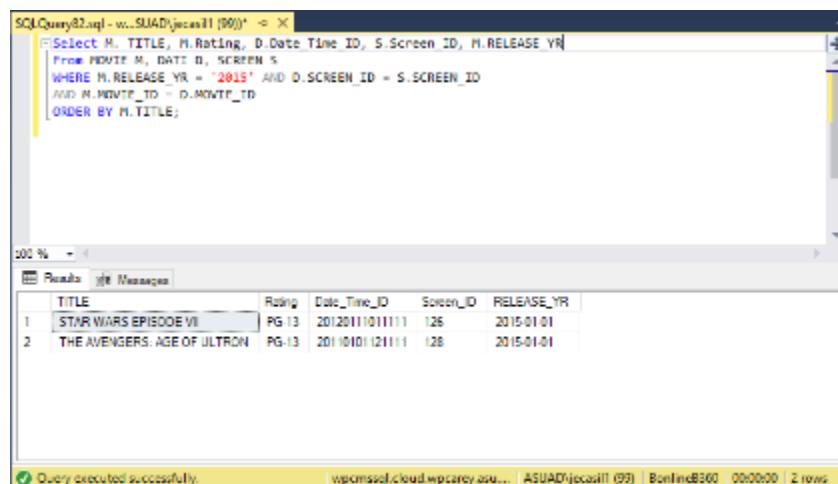
```
VALUES('600028', '2', '4.00', '8.00', '620');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600029', '1', '4.00', '4.00', '601');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600030', '1', '4.00', '4.00', '630');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600031', '5', '4.00', '20.00', '640');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600032', '1', '5.00', '5.00', '550');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600033', '3', '5.00', '15.00', '551');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600034', '1', '0.50', '0.50', '552');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600035', '2', '0.50', '1.00', '553');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600036', '1', '1.50', '1.50', '530');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600037', '4', '1.50', '6.00', '531');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600038', '1', '0.50', '0.50', '532');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600039', '2', '0.50', '1.00', '533');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600040', '1', '2.00', '2.00', '500');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600041', '1', '5.00', '5.00', '510');
INSERT INTO ORDERS (ORDER_ID, QUANTITY, COST, EXT_COST, MENU_ITEM_ID)
VALUES('600042', '1', '10.00', '10.00', '520');
```

# TESTS AND QUERIES

## (SCRIPTS & SCREEN CAPTURE)

What is the name of the movies that were released in 2015? List the rating, date, time, and screen number.

Figure 6.1



The screenshot shows a SQL Server Management Studio (SSMS) interface. The query window displays the following SQL code:

```
Select M.Title, M.Rating, D.Date_Time_ID, S.Screen_ID, M.RELEASE_YR
From MOVIE M, DATE D, SCREEN S
WHERE M.RELEASE_YR = '2015' AND D.SCREEN_ID = S.SCREEN_ID
AND M.MOVIE_ID = D.MOVIE_ID
ORDER BY M.Title;
```

The results window shows the following data:

	TITLE	Rating	Date_Time_ID	Screen_ID	RELEASE_YR
1	STAR WARS EPISODE VII	PG 13	2012011011111	128	20150101
2	THE AVENGERS: AGE OF ULTRON	PG-13	20110101121111	128	20150101

At the bottom of the results window, it says "Query executed successfully." and shows the session details: wpcmssd.cloud.wpcarey.asu... | ASUAD\jccasill (93) | BonfireDB300 | 00:00:00 | 2 rows.

Write a join statement to get the employee's last name and theater address. Include the city, state, and order the result by the zip code. Store the result in Employee\_Location

SQLQuery88.sql - w...UAD\jecasil1 (108)\*

```
SELECT *
FROM EMPLOYEE_LOCATION
```

100 %

Results Messages

	EMPLOYEE LAST NAME	THEATRE ADDRESS	THEATRE CITY	THEATRE STATE	THEATRE ZIP
1	BUBBLES	1912 S. APRIL AVE.	PHOENIX	AZ	85001
2	GRAHAM	1912 S. APRIL AVE.	PHOENIX	AZ	85001
3	IRELAND	1912 S. APRIL AVE.	PHOENIX	AZ	85001
4	MOOSE	1912 S. APRIL AVE.	PHOENIX	AZ	85001
5	NEWT	1912 S. APRIL AVE.	PHOENIX	AZ	85001
6	SANDERS	1912 S. APRIL AVE.	PHOENIX	AZ	85001
7	LUND	52577 N. LUCAS LN.	SCOTTSDALE	AZ	85054
8	SCARN	52577 N. LUCAS LN.	SCOTTSDALE	AZ	85054
9	SPENCER	52577 N. LUCAS LN.	SCOTTSDALE	AZ	85054
10	WALKER	52577 N. LUCAS LN.	SCOTTSDALE	AZ	85054
11	CHRISTIANSEN	20126 E. ASSEMBL...	QUEEN CREEK	AZ	85140
12	MCARTHUR	20126 E. ASSEMBL...	QUEEN CREEK	AZ	85140
13	PRIOR	20126 E. ASSEMBL...	QUEEN CREEK	AZ	85140
14	BERRY	19708 N. ADRIAN A...	GILBERT	AZ	85142
15	JONES	19708 N. ADRIAN A...	GILBERT	AZ	85142
16	REES	15 W. PREHISTORI...	MESA	AZ	85201
17	ROKS	15 W. PREHISTORI...	MESA	AZ	85201
18	STRONG	15 W. PREHISTORI...	MESA	AZ	85201
19	CARLSON	85515 W. MCFLY RD.	TEMPE	AZ	85202
20	FOX	85515 W. MCFLY RD.	TEMPE	AZ	85202
21	MOSS	85515 W. MCFLY RD.	TEMPE	AZ	85202
22	OWENS	85515 W. MCFLY RD.	TEMPE	AZ	85202
23	COOK	111401 W. PRINET ...	CHANDLER	AZ	85224
24	DESPAIN	111401 W. PRINET ...	CHANDLER	AZ	85224
25	LION	111401 W. PRINET ...	CHANDLER	AZ	85224
26	GARRETT	62075 S. SPIELBER...	FOUNTAIN H...	AZ	85269
27	JAMES	62075 S. SPIELBER...	FOUNTAIN H...	AZ	85269
28	MONSON	62075 S. SPIELBER...	FOUNTAIN H...	AZ	85269
29	BLUE	1954 E. PRIDE RD...	PEORIA	AZ	85301
30	JOHNSON	71390 N. SPECTER...	GLENDALE	AZ	85301

Query executed successfully.

wpcmssql.cloud.wpcarey.esu... ASUAD\jecasil1 (108) BonlineB360 00:00:00 | 40 rows

SQLQuery87.sql - w...SUAD\jecasil1 (98)\*

```
CREATE VIEW EMPLOYEE_LOCATION AS
SELECT E.L_NAME AS 'EMPLOYEE LAST NAME', T.ADDRESS AS 'THEATRE ADDRESS',
T.CITY AS 'THEATRE CITY', T.STATE AS 'THEATRE STATE', T.ZIP AS 'THEATRE ZIP'
FROM EMPLOYEE E, THEATRE T
WHERE E.THEATRE_ID = T.THEATRE_ID
GROUP BY T.ZIP, E.L_NAME, T.ADDRESS, T.CITY, T.STATE;
```

100 %

Messages

Commands completed successfully.

Completion time: 2022-04-17T14:08:28.2703796-07:00

100 %

Query executed successfully.

wpcmssql.cloud.wpcarey.esu... ASUAD\jecasil1 (98) BonlineB360 00:00:00 | 0 rows

/\* Create a View that shows which employees have earned the smallest tips? Show only the tips that are \$5 or less and which employee (by First & Last Name, last name first) earned the tip? Rename tip as "Tips Less than \$5" and list the total lowest to highest. Place the result in Tips\_Less\_Than\_\$5. \*/

```

create view Tips_Less_Than_$5 as
select i.Tip as 'Tips Less Than $5', e.L_Name as 'Last Name', e.F_Name as 'First Name'
from Invoice i, Employed e
where i.Tip <= 5 and i.Employed_ID = e.Employed_ID
group by i.Tip, e.L_Name, e.F_Name

```

The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure with nodes like blottail, bmshaw1, bnegede, BonlineB360, and various tables (CUSTOMER, DATI, EMPLOYED, INVOICE, MENU, MOVIE, ORDERS, SCREEN, THEATRE).
- SQL Query Editor:** Contains the T-SQL code for creating the view.
- Results Grid:** Displays the output of the query, showing columns: Tip Less Than \$5, Last Name, and First Name. The data includes rows for BUBBLES, FOX, GRAHAM, OWENS, SPENCER, WALKER, CHRISTIANSEN, RIGGS, VICTOR, LION, RICKS, MILLER, DESPAIN, TAX, SANDERS, SCARN, COOK, LEE, MCARTHUR, MONSON, MOOSE, BALLARD, BERRY, DESTINY, PRIOR, STRONG, LION, and MOSS.
- Status Bar:** Shows "Query executed successfully." and other session details.

	Tips Less Than \$5	Last Name	First Name
1	0.00	BUBBLES	FRED
2	0.00	FOX	MIA
3	0.00	GRAHAM	HARRY
4	0.00	OWENS	ANNIE
5	0.00	SPENCER	SUNNY
6	0.20	WALKER	CHARLIE
7	0.80	CHRISTIANSEN	KATIE
8	0.80	RIGGS	GREG
9	0.80	VICTOR	CRISTY
10	1.00	LION	CARRIE
11	1.00	RICKS	STELLA
12	1.10	MILLER	JACOB
13	1.20	DESPAIN	LANE
14	1.20	JOHNSON	ELI
15	1.20	SANDERS	HALEY
16	1.20	SCARN	MICHAEL
17	1.40	COOK	LILLY
18	1.40	LEE	JANE
19	1.40	MCARTHUR	POLLY
20	1.40	MONSON	ARCHER
21	1.40	MOOSE	ELLE
22	1.60	BALLARD	NINA
23	1.60	BERRY	KATYA
24	1.60	DESTINY	ERIN
25	1.60	PRIOR	ANNE
26	1.60	STRONG	UMA
27	2.00	LION	CARRIE
28	2.00	MOSS	PHOEBE

```
/* For each check, display menu items sold, total check amount and tips added. */
```

```

select i.Check_ID, m.Item_Desc, i.Total, I.Tip
from Invoice i, Orders o, Menu m
where i.Order_ID = o.Order_ID and o.Menu_Item_ID = m.Menu_Item_ID

```

check_id	ITEM_DESC	TOTAL	TIP
1	200001 POPCORN	12.00	2.40
2	200002 POPCORN	7.00	1.40
3	200003 POPCORN	8.00	1.60
4	200004 POPCORN	27.00	5.40
5	200005 SODA	5.50	1.10
6	200006 SODA	6.00	1.20
7	200007 SODA	13.00	2.60
8	200008 SODA	7.00	1.40
9	200009 SODA	15.00	3.00
10	200010 MOVIE TICKET	11.00	2.20
11	200011 MOVIE TICKET	28.00	5.60
12	200012 MOVIE TICKET	12.00	2.40
13	200013 MOVIE TICKET	39.00	7.80
14	200014 CHEESE PIZZA	7.00	1.40
15	200015 CHEESE PIZZA	7.50	1.20
16	200016 CHEESE PIZZA	16.00	3.20
17	200017 VEGGIE PIZZA	7.00	1.40
18	200018 VEGGIE PIZZA	7.50	1.20
19	200019 VEGGIE PIZZA	8.00	1.60
20	200020 PEPPERONI PIZZA	21.00	4.20
21	200021 PEPPERONI PIZZA	30.00	6.00
22	200022 PEPPERONI PIZZA	8.00	1.60
23	200023 MEAT LOVERS PIZZA	7.00	1.40
24	200024 MEAT LOVERS PIZZA	30.00	6.00
25	200025 MEAT LOVERS PIZZA	8.00	1.60
26	200026 M&Ms	12.00	2.40
27	200027 SKITTLES	4.00	0.80

Query executed successfully.

```
/* Create a View that shows which employees have earned the smallest tips? Show only the tips
that are $5 or less and which employee (by First & Last Name,
last name first) earned the tip? Rename tip as "Tips Less than $5" and list the total lowest to
highest. Place the result in Tips_Less_Than_$5. */
```

```
SELECT MAX(TOTAL) AS [TOTAL FOR LAST NAME STARTING WITH J], L_NAME
FROM INVOICE, CUSTOMER
WHERE TOTAL IN
(SELECT TOTAL
FROM INVOICE
WHERE TICKET_ID IN
(SELECT TICKET_ID
FROM TICKET
WHERE CUSTOMER_ID IN
(SELECT CUSTOMER_ID
FROM CUSTOMER
WHERE L_NAME LIKE 'J%')))
GROUP BY L_NAME, TOTAL
ORDER BY TOTAL DESC;
```

\*produced no data\*

The screenshot shows a SQL Server Management Studio window with two tabs: 'SQLQuery45.sql' and 'SQLQuery44.sql'. The code in 'SQLQuery45.sql' is a nested query that selects the maximum total amount for customers whose last name starts with 'J'. The code is as follows:

```

SELECT MAX(TOTAL) AS [TOTAL FOR LAST NAME STARTING WITH J], L_NAME
FROM INVOICE, CUSTOMER
WHERE TOTAL IN
(SELECT TOTAL
FROM INVOICE
WHERE TICKET_ID IN
(SELECT TICKET_ID
FROM TICKET
WHERE CUSTOMER_ID IN
(SELECT CUSTOMER_ID
FROM CUSTOMER
WHERE L_NAME LIKE 'J%')))
GROUP BY L_NAME, TOTAL
ORDER BY TOTAL DESC;

```

The results pane shows a single row with the header 'TOTAL FOR LAST NAME STARTING WITH J | L\_NAME |'. The data row is empty.

Deleting a theatre with an employee assigned should fail

The screenshot shows a SQL Server Management Studio window with a single tab labeled 'Messages'. The message is as follows:

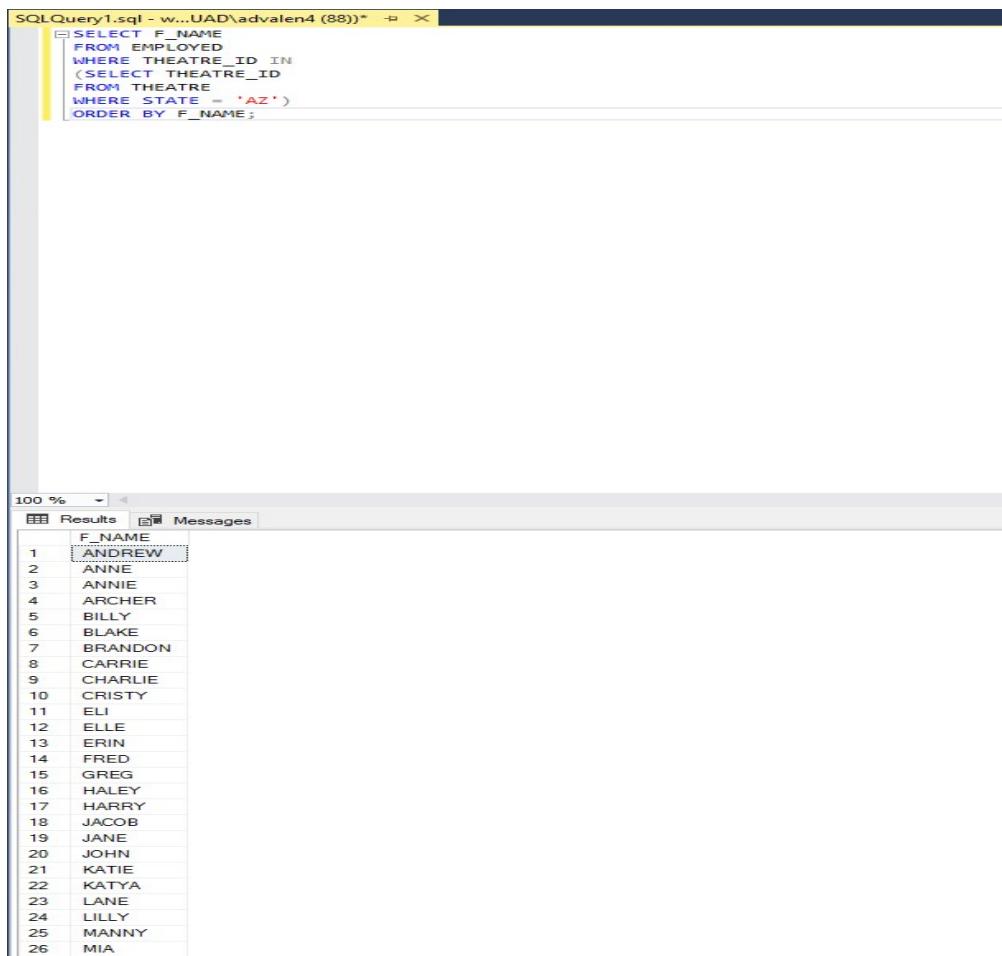
```

Msg 547, Level 16, State 0, Line 1
The DELETE statement conflicted with the REFERENCE constraint "FK__EMPLOYED__theatre_id". The conflict occurred in database "BonlineB360", table "dbo.EMPLOYED", column 'theatre_id'.
The statement has been terminated.
Completion time: 2022-04-17T11:37:19.4822412+09:00

```

What the first name of employees that are related to the state of Arizona? List the first name alphabetically.

```
SELECT F_NAME  
FROM EMPLOYED  
WHERE THEATRE_ID IN  
(SELECT THEATRE_ID  
FROM THEATRE  
WHERE STATE = 'AZ')  
ORDER BY F_NAME;
```



The screenshot shows a SQL query window titled "SQLQuery1.sql - w...UAD\advalen4 (88)\*". The query is:

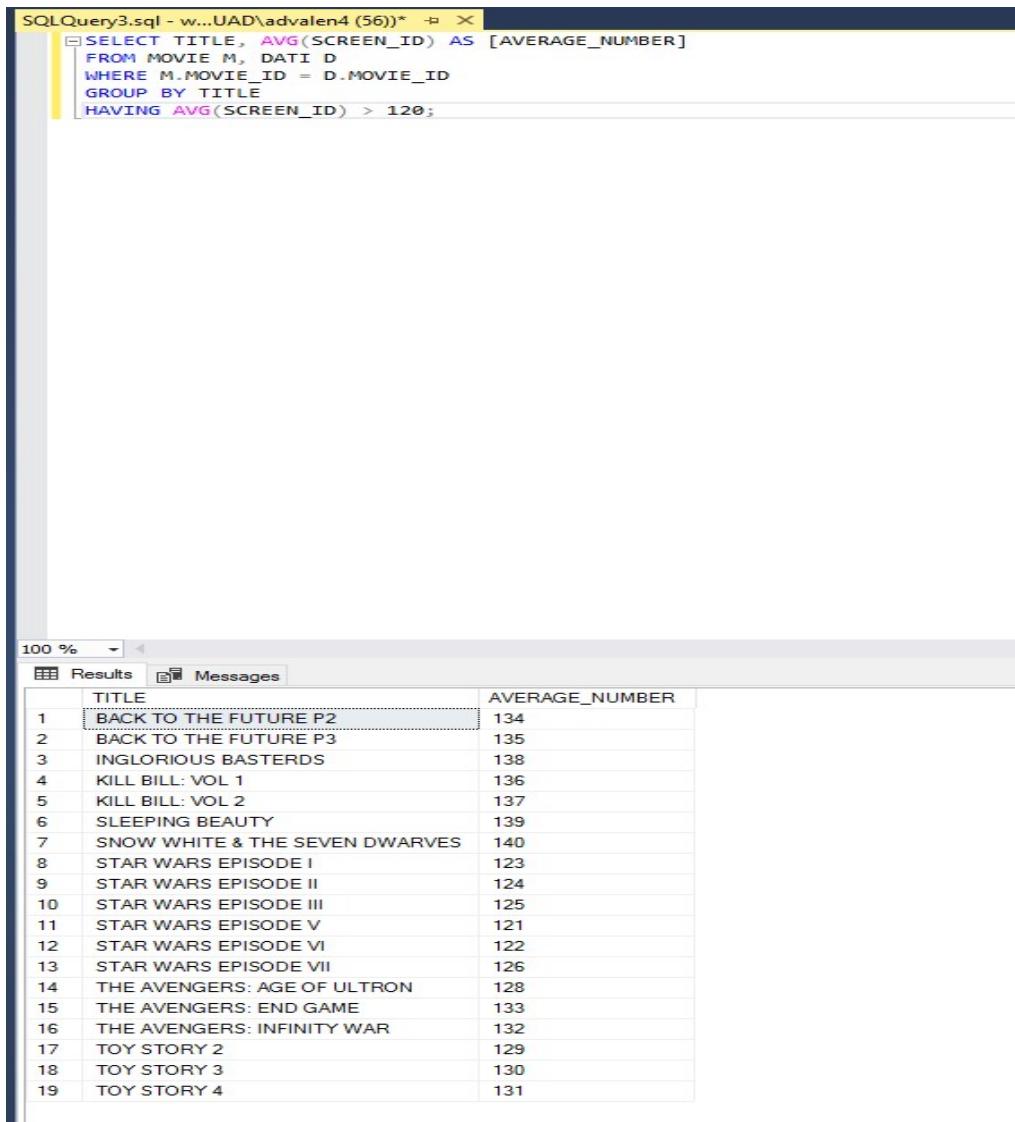
```
SELECT F_NAME  
FROM EMPLOYED  
WHERE THEATRE_ID IN  
(SELECT THEATRE_ID  
FROM THEATRE  
WHERE STATE = 'AZ')  
ORDER BY F_NAME;
```

The results grid displays 26 rows of employee first names, each preceded by a number from 1 to 26. The first few rows are:

	F_NAME
1	ANDREW
2	ANNE
3	ANNIE
4	ARCHER
5	BILLY
6	BLAKE
7	BRANDON
8	CARRIE
9	CHARLIE
10	CRISTY
11	ELI
12	ELLE
13	ERIN
14	FRED
15	GREG
16	HALEY
17	HARRY
18	JACOB
19	JANE
20	JOHN
21	KATIE
22	KATYA
23	LANE
24	LILLY
25	MANNY
26	MIA

What is the average screen number for each movie? Include the title and have only the screen numbers over 120.

```
SELECT TITLE, AVG(SCREEN_ID) AS [AVERAGE_NUMBER]
FROM MOVIE M, DATI D
WHERE M.MOVIE_ID = D.MOVIE_ID
GROUP BY TITLE
HAVING AVG(SCREEN_ID) > 120;
```



The screenshot shows a SQL Server Management Studio window with a query editor and a results grid. The query editor contains the following T-SQL code:

```
SQLQuery3.sql - w...UAD\advalen4 (56)* -> X
SELECT TITLE, AVG(SCREEN_ID) AS [AVERAGE_NUMBER]
FROM MOVIE M, DATI D
WHERE M.MOVIE_ID = D.MOVIE_ID
GROUP BY TITLE
HAVING AVG(SCREEN_ID) > 120;
```

The results grid displays the following data:

	TITLE	AVERAGE_NUMBER
1	BACK TO THE FUTURE P2	134
2	BACK TO THE FUTURE P3	135
3	INGLORIOUS BASTERDS	138
4	KILL BILL: VOL 1	136
5	KILL BILL: VOL 2	137
6	SLEEPING BEAUTY	139
7	SNOW WHITE & THE SEVEN DWARVES	140
8	STAR WARS EPISODE I	123
9	STAR WARS EPISODE II	124
10	STAR WARS EPISODE III	125
11	STAR WARS EPISODE V	121
12	STAR WARS EPISODE VI	122
13	STAR WARS EPISODE VII	126
14	THE AVENGERS: AGE OF ULTRON	128
15	THE AVENGERS: END GAME	133
16	THE AVENGERS: INFINITY WAR	132
17	TOY STORY 2	129
18	TOY STORY 3	130
19	TOY STORY 4	131

Which menu item sells best, has highest sales? Which bar item?

```
SELECT ITEM_DESC, TOTAL  
FROM MENU,INVOICE  
ORDER BY TOTAL DESC;
```

SQLQuery3.sql - w...UAD\advalen4 (56)\* - X

```
SELECT ITEM_DESC AS [BAR_ITEM], TOTAL
FROM MENU,INVOICE
ORDER BY TOTAL DESC;
```

100 % < Results Messages

	BAR_ITEM	TOTAL
1	MOVIE TICKET	39.00
2	MOVIE TICKET	39.00
3	MOVIE TICKET	39.00
4	MOVIE TICKET	39.00
5	BEER	39.00
6	WINE	39.00
7	COCKTAIL	39.00
8	VEGGIE STICKS W/ DRESSING	39.00
9	VEGGIE STICKS W/ HUMMUS	39.00
10	SIDE OF DRESSING	39.00
11	SIDE OF HUMMUS	39.00
12	SOFT PRETZEL BITES W/ CHEESE	39.00
13	SOFT PREZTEL BITES W/ MUSTARD	39.00
14	SIDE OF CHEESE	39.00
15	SIDE OF MUSTARD	39.00
16	M&MS	39.00
17	PEANUT M&MS	39.00
18	SKITTLES	39.00
19	REDVINES	39.00
20	JUNIOR MINTS	39.00
21	GOOD & PLENTY	39.00
22	CHEESE PIZZA	39.00
23	CHEESE PIZZA	39.00
24	CHEESE PIZZA	39.00
25	VEGGIE PIZZA	39.00
26	VEGGIE PIZZA	39.00

What movie is playing in the theater that has employees whose First name is John? Include the phone number, address, and state. List the zip alphabetically.

Figure 6.2

```
SELECT E.F_NAME, T.THEATRE_ID, E.PHN_NUM, E.ADDRESS, E.STATE, T.NAME,  
T.ZIP  
FROM EMPLOYED E, THEATRE T  
WHERE E.F_NAME = 'JOHN' and E.THEATRE_ID = T.THEATRE_ID  
ORDER BY [ZIP]
```

The screenshot shows the SQL Server Management Studio interface. In the top pane, there are three tabs: 'SQLQuery43.sql - w...SUAD\ejacks4 (76)\*' (highlighted in yellow), 'SQLQuery41.sql - not connected\*', and 'SQLQuery39.sql - not connected'. The yellow-highlighted tab contains the SQL query:

```
SELECT E.F_NAME, T.THEATRE_ID, E.PHN_NUM, E.ADDRESS, E.STATE, T.NAME, T.ZIP  
FROM EMPLOYED E, THEATRE T  
WHERE E.F_NAME = 'JOHN' and E.THEATRE_ID = T.THEATRE_ID  
ORDER BY [ZIP]
```

In the bottom pane, there are two tabs: 'Results' (selected) and 'Messages'. The 'Results' tab displays the query results in a table:

	F_NAME	THEATRE_ID	PHN_NUM	ADDRESS	STATE	NAME	ZIP
1	JOHN	101	6029980533	1534 E 34TH ST.	AZ	TITANIC	85001

The 'Messages' tab is empty. At the bottom of the interface, a green status bar says 'Query executed successfully.' and 'wpc'.

Two INSERT statements in a row – the first one adding a new Customer, the second one adding a new ticket for that Customer, should succeed.

Figure 6.3

```
INSERT INTO CUSTOMER
VALUES('000060', 'JACKSON', 'ELIJAH', 'EJAC@ICLOUD.COM', '2096845149','2001-12-05',
'2022-04-16', '000003', 'OSCAR')
INSERT INTO TICKET
VALUES('100041', '20121205083501', 'A10008', '000001', '200041', '000060')

SQLQuery46.sql - w...SUAD\ejacks4 (80)*      SQLQuery56.sql - w...SUAD\ejacks4 (84)*      SQLQuery53.sql - w...SUAD\ejacks4 (69)*      SQLQuery59.
[+] INSERT INTO CUSTOMER
[+]   VALUES('000060', 'JACKSON', 'ELIJAH', 'EJAC@ICLOUD.COM', '2096845149', '2001-12-05', '2022-04-16', '000003', 'OSCAR')
[+] INSERT INTO TICKET
[+]   VALUES('100041', '20121205083501', 'A10008', '000001', '200041', '000060') |
```

100 % < Messages

(1 row affected)

(1 row affected)

Completion time: 2022-04-16T02:31:29.9769561-07:00

How many customers watched the movie "Toy Story" in 2012? Include the customer first and last names. List the first name alphabetically.

Figure 6.4

```
SELECT F_NAME, L_NAME  
FROM CUSTOMER  
WHERE CUSTOMER_ID = (SELECT CUSTOMER_ID  
FROM TICKET WHERE DATE_TIME_ID BETWEEN '2012010101' AND '2012123199'  
AND MOVIE_ID = 'A10009')  
ORDER BY F_NAME ASC
```

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left is the Object Explorer pane, which lists the database 'BonlineB360' and its tables: CUSTOMER, DATI, EMPLOYED, INVOICE, MENU, and MOVIE. The MOVIE table is expanded to show columns: MOVIE\_ID (PK), TITLE, RATING, and RELEASE\_YR. The central pane contains a query window with the following SQL code:

```
SELECT F_NAME, L_NAME
FROM CUSTOMER
WHERE CUSTOMER_ID = (SELECT CUSTOMER_ID
FROM TICKET WHERE DATE_TIME_ID BETWEEN '2012010101' AND '2012123199'
AND MOVIE_ID = 'A10009')
ORDER BY F_NAME ASC
```

The results pane below the query window shows a single row of data:

F_NAME	L_NAME
CARL	RED

At the bottom of the SSMS window, a status bar indicates: "Query executed successfully." and "wpcmssql.cloud.wpcarey.asu.edu | ASUAD\ejmoss1 (67) | BonlineB360 | 00:00:00 | 1 rows".

What are the title of all movies that were released in 2010? List the movie titles Unalphabetically.

```
SELECT TITLE FROM MOVIE WHERE RELEASE_YR='2010'
```

Figure 6.5

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The Object Explorer on the left shows a database named 'BonlineB360' containing several tables like CUSTOMER, DATA, EMPLOYED, INVOICE, MENU, and MOVIE. The central pane displays a query window with the following content:

```
SELECT TITLE FROM MOVIE WHERE RELEASE_YR='2010'
```

The results pane below shows the output of the query:

TITLE
HARRY POTTER & DEATHLY HALLOWS PT
TOY STORY 3

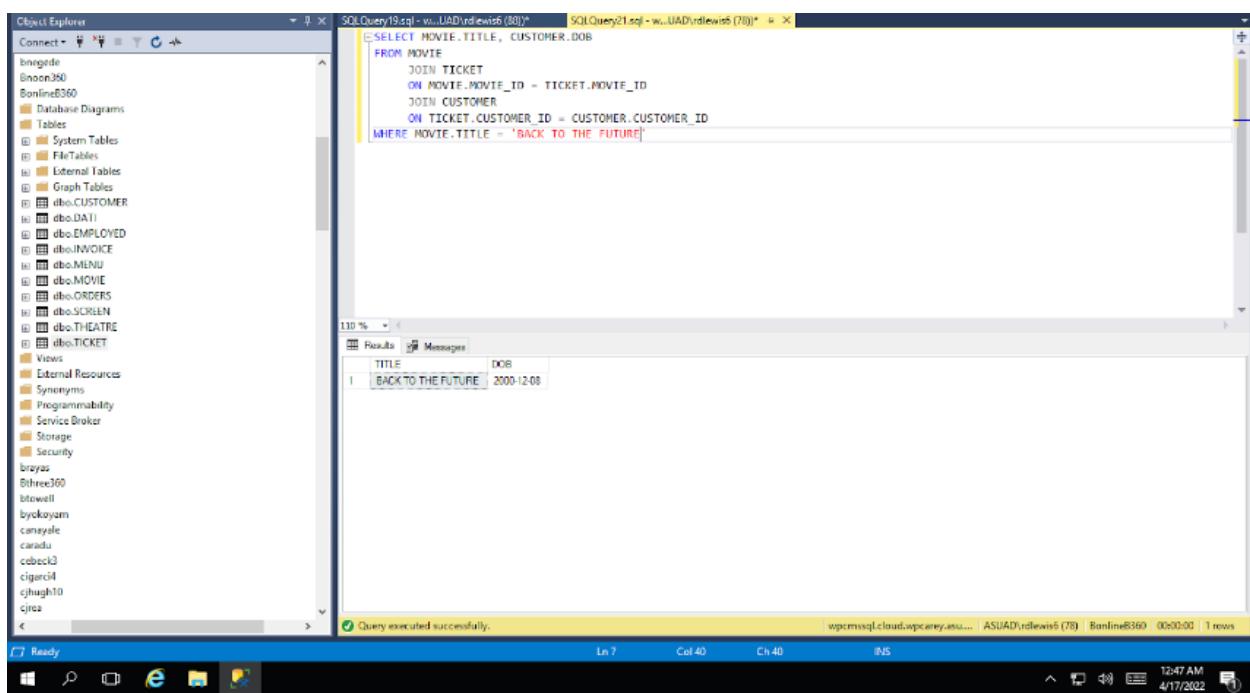
At the bottom of the results pane, a message indicates: "Query executed successfully." and shows the status bar with "2 rows".

**What is the date of birth in the years 1990-1995 of customers who watched "Back to the future"?**

None, there was only one customer who watched 'Back to the future' and the DOB was 2000-12-08

```
SELECT MOVIE.TITLE, CUSTOMER.DOB  
FROM MOVIE  
JOIN TICKET  
ON MOVIE.MOVIE_ID = TICKET.MOVIE_ID  
JOIN CUSTOMER  
ON TICKET.CUSTOMER_ID = CUSTOMER.CUSTOMER_ID  
WHERE MOVIE.TITLE = 'BACK TO THE FUTURE'
```

Figure 6.6



A similar two INSERT statements executed in the opposite order should fail.

Figure 6.7

The screenshot shows the Object Explorer on the left and a query window on the right. The query window contains two INSERT statements:

```
INSERT
VALUES('288866', 'SNACK STAND', '32.00', '0.00', '1.72', '6.48', '888866', '20110101121111', '688866')
INTO INVOICE;

INSERT
VALUES('200067', 'SNACK STAND', '37.00', '0.00', '1.42', '7.40', '000067', '20110304040454', '600067')
INTO INVOICE;
```

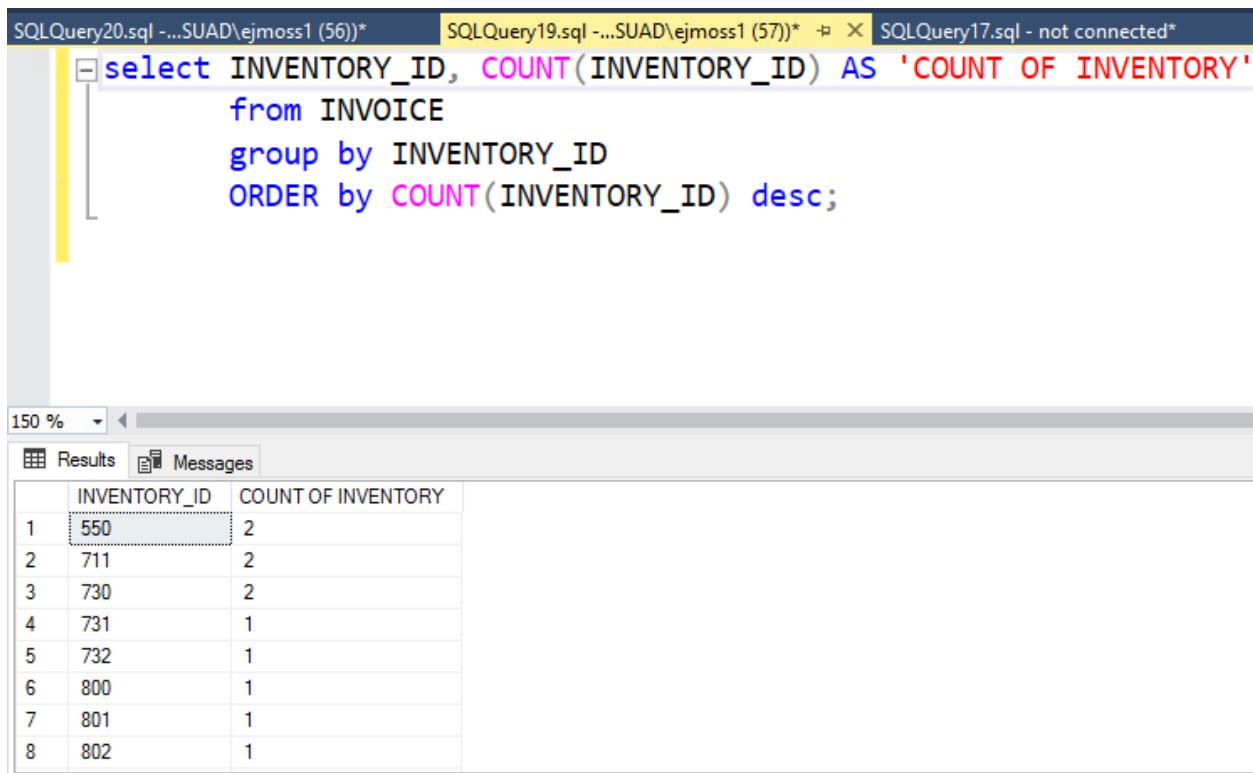
The status bar at the bottom indicates "Query completed with errors." The error message in the Messages pane is:

Msg 156, Level 15, State 1, Line 2  
Incorrect syntax near the keyword 'VALUES'.

Completion time: 2022-04-17T10:48:57.555729+09:00

Query: Most popular inventory item

Figure 6.8



The screenshot shows a SQL Server Management Studio window with three tabs at the top: 'SQLQuery20.sql - ...SUAD\ejmoss1 (56)\*' (disabled), 'SQLQuery19.sql - ...SUAD\ejmoss1 (57)\*' (selected), and 'SQLQuery17.sql - not connected\*'. The main area contains a T-SQL query:

```
select INVENTORY_ID, COUNT(INVENTORY_ID) AS 'COUNT OF INVENTORY'
from INVOICE
group by INVENTORY_ID
ORDER by COUNT(INVENTORY_ID) desc;
```

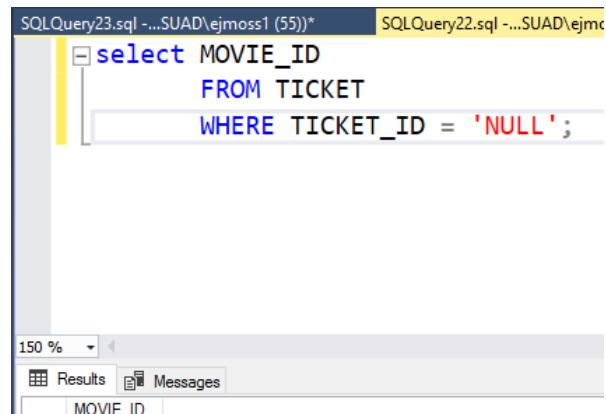
The 'Results' tab is selected, displaying the following table:

	INVENTORY_ID	COUNT OF INVENTORY
1	550	2
2	711	2
3	730	2
4	731	1
5	732	1
6	800	1
7	801	1
8	802	1

Answer: The most popular inventory items are 550 (soft pretzel bites with cheese), 711 (10in veggie pizza), and 730 (8in meat lovers pizza)

Query: Movies that did not sell any tickets

Figure 6.9



The screenshot shows a SQL Server Management Studio interface. At the top, there are two tabs: 'SQLQuery23.sql - ...SUAD\ejmoss1 (55)\*' and 'SQLQuery22.sql - ...SUAD\ejm...'. Below the tabs, a query window contains the following SQL code:

```
select MOVIE_ID  
      FROM TICKET  
     WHERE TICKET_ID = 'NULL';
```

Below the query window is a results pane with a header row containing the column name 'MOVIE\_ID'. The main body of the results pane is currently empty, indicating no data was returned.

Answer: All movies sold tickets

Figure 6.10

Query: Which movie sold the most tickets?

```
SQLQuery3.sql - w...SUAD\cebeck3 (76)* ⇢ X SQLQuery2.sql - w...SUAD\
    select * from Movie_Sales order by 2 desc |
```

100 %

	Movie Title	Number of Tickets
1	HARRY POTTER & DEATHLY HALLOWS P1	4
2	STAR WARS EPISODE VIII	4
3	TITANIC	3
4	HARRY POTTER & THE SORCERERS STONE	2
5	INGLORIOUS BASTERDS	1
6	JAWS	1
7	JURASSIC PARK	1
8	KILL BILL: VOL 1	1
9	LION KING	1
10	ROCKY	1
11	SLEEPING BEAUTY	1
12	SNOW WHITE & THE SEVEN DWARVES	1
13	STAR WARS	1
14	STAR WARS EPISODE I	1
15	STAR WARS EPISODE II	1
16	STAR WARS EPISODE III	1
17	STAR WARS EPISODE IX	1
18	STAR WARS EPISODE VI	1
19	STAR WARS EPISODE VII	1
20	TOY STORY	1
21	TOY STORY 2	1
22	TOY STORY 3	1
23	TOY STORY 4	1
24	THE AVENGERS	1
25	THE AVENGERS: AGE OF ULTRON	1
26	THE AVENGERS: END GAME	1
27	THE AVENGERS: INFINITY WAR	1
28	HARRY POTTER & DEATHLY HALLOWS P2	1
29	HARRY POTTER & GOBLET OF FIRE	1
30	HARRY POTTER & HALF BLOOD PRINCE	1
31	HARRY POTTER & ORDER OF THE PHOENIX	1

Query executed successfully.

Answer: The two movies that tied for most tickets sold were “Harry Potter and Deathly Hallows Pt. 1” and “Star Wars Episode VIII”

Figure 6.11

Query/view: Names of the customers who have achieved EGOT status

```

SQLQuery1.sql - wp...UAD\cigarc4 (66)* ✎ X [SQL]
[U] CREATE VIEW CUSTOMERLEVEL AS
    SELECT RWD_LVL_ID, CUSTOMER_ID, F_NAME
    FROM CUSTOMER;

SQLQuery2.sql - wp...UAD\cigarc4 (67)* ✎ X [SQL]
SELECT F_NAME
FROM CUSTOMERLEVEL
WHERE RWD_LVL_ID = 'EGOT';

```

80 %

F_NAME

Answer: No customers have achieved the EGOT status yet.

Figure 6.12

Query: Name of customer with the most visits

```

SQLQuery4.sql - w...SUAD\cebeck3 (60)* ✎ X SQLQuery3.sql - w...SU
select CUSTOMER_ID, F_NAME, L_NAME, NUM_VISIT
from CUSTOMER
order by NUM_VISIT desc

```

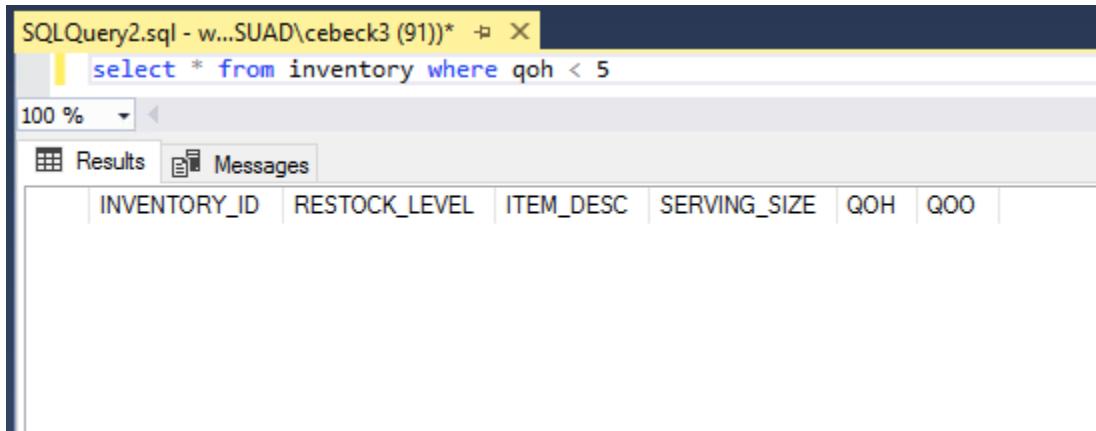
100 %

	CUSTOMER_ID	F_NAME	L_NAME	NUM_VISIT
1	000012	LUIS	CERVANTEZ	5
2	000006	ANDREW	LEE	000031
3	000004	LUCY	WELLS	000017
4	000007	SEAN	PARKER	000016
5	000013	MAX	KITCHEN	000015
6	000020	DANIELA	RUIZ	000015
7	000021	DANNY	RUIZ	000015
8	000010	ADAM	RUIZ	000015
9	000031	BRANDON	SMITH	000015
10	000032	ROBERT	SMART	000015
11	000033	ZABDIEL	SANTOS	000015
12	100078	NAMER	ERIC	000015
13	000036	SARAH	VALDEZ	000009

Answer: Andrew Lee has the most visits because he has visited the theaters 31 times. The top result only had 5 visits but appears first because the zeroes were not added when a different query was run.

Query: Items for which Quantity on Hand (QoH) that is less than 5.

Figure 6.13

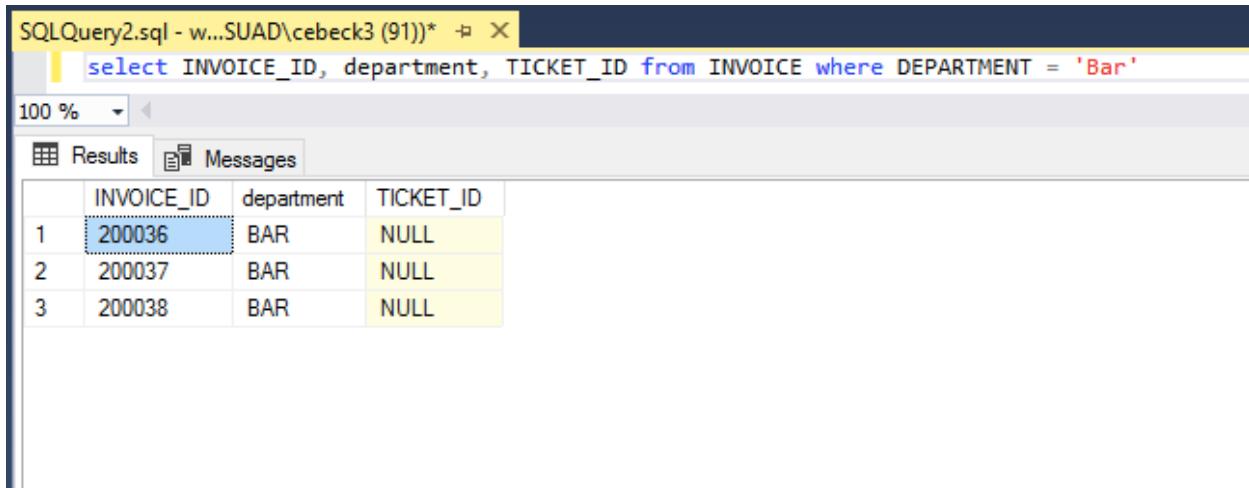


The screenshot shows a SQL Server Management Studio window titled "SQLQuery2.sql - w...SUAD\cebeck3 (91)\*". The query is "select \* from inventory where qoh < 5". The results grid has columns: INVENTORY\_ID, RESTOCK\_LEVEL, ITEM\_DESC, SERVING\_SIZE, QOH, and QOO. There are no rows in the grid.

Answer: No Items have a Quantity on Hand (QOH) that is less than 5.

Query: Total Customer Visits that were only for the Bar.

Figure 6.14



The screenshot shows a SQL Server Management Studio window titled "SQLQuery2.sql - w...SUAD\cebeck3 (91)\*". The query is "select INVOICE\_ID, department, TICKET\_ID from INVOICE where DEPARTMENT = 'Bar'". The results grid has columns: INVOICE\_ID, department, and TICKET\_ID. There are three rows with values: (1, BAR, NULL), (2, BAR, NULL), and (3, BAR, NULL). The INVOICE\_ID column is bolded.

Answer: There were three invoices that were only for the Bar (invoice # 200036, 200037, 200038).

# TRIGGERS AND PROCEDURES

## (SCRIPTS & SCREEN CAPTURE)

Trigger Script for Inventory Reduction when a New Invoice is added:

```
create trigger REFINV on INVOICE
after insert as
update INVENTORY
set INVENTORY.QOH = INVENTORY.QOH - MENU.QUANTITY
from MENU, INVOICE
where MENU.MENU_ID = INVOICE.MENU_ID and
INVOICE.INVENTORY_ID = INVENTORY.INVENTORY_ID;
```

Figure 7.1

```
CREATE TRIGGER REFINV ON INVOICE
AFTER INSERT AS
UPDATE INVENTORY
SET INVENTORY.QOH = INVENTORY.QOH - MENU.QUANTITY
FROM MENU, INVOICE
WHERE MENU.MENU_ID = INVOICE.MENU_ID AND INVOICE.INVENTORY_ID = INVENTORY.INVENTORY_ID
```

100 % < Messages  
Commands completed successfully.  
Completion time: 2022-04-23T18:46:36.0481354-07:00

100 % < Query executed successfully. | wpcmssql.cloud.wpcarey.asu... | ASUAD\jecasil1 (70) | BonlineB360 | 00:00

Successful Creation of Inventory Reduction Trigger

Figure 7.2

SQLQuery7.sql - wp...UAD\jecasil1 (100)\* SQLQuery6.sql - wp...UAD\jecasil1 (109)\* SQLQuery5.sql - wp...SUAD\jecasil1 (31)

```
SELECT *
FROM INVENTORY
```

100 %

Results Messages

	INVENTORY_ID	RESTOCK_LEVEL	ITEM_DESC	SERVING_SIZE	QOH	QOO
15	590	500	HAMBURGER W/ CHEESE	NULL	100	0
16	595	250	VEGGIE BURGER W/ CHEESE	NULL	248	0
17	600	500	M&MS	NULL	44	0
18	601	500	PEANUT M&MS	NULL	233	100
19	610	300	SKITTLES	NULL	461	0
20	620	300	REDVINES	NULL	0	0
21	630	200	JUNIOR MINTS	NULL	393	1000
22	640	100	GOOD & PLENTY	NULL	44	100
23	700	100	CHEESE PIZZA	8 IN	60	150
24	701	50	CHEESE PIZZA	10 IN	160	0
25	702	100	CHEESE PIZZA	14 IN	22	200
26	710	100	VEGGIE PIZZA	8 IN	6	150
27	711	50	VEGGIE PIZZA	10 IN	60	0
28	712	100	VEGGIE PIZZA	14 IN	0	200
29	720	100	PEPPERONI PIZZA	8 IN	6	250
30	721	50	PEPPERONI PIZZA	10 IN	0	100
31	722	100	PEPPERONI PIZZA	14 IN	120	0
32	730	100	MEAT LOVERS PIZZA	8 IN	101	0
33	731	50	MEAT LOVERS PIZZA	10 IN	0	0
34	732	100	MEAT LOVERS PIZZA	14 IN	120	0
35	800	700	POPCORN	SMALL	689	0
36	801	700	POPCORN	MED	689	0
37	802	700	POPCORN	LARG	689	0
38	803	700	POPCORN	XL	689	0
39	900	500	SODA	SMALL	412	0
40	901	500	SODA	MED	501	0
41	903	500	SODA	LARG	920	0
42	904	500	SODA	XL	880	0

Query executed successfully.

wpcmssql.cloud

Inventory Table before creating new Invoice to test REFINV Trigger

REFINV Trigger Test Invoice Script:

```
insert INVOICE values ('200047', 'SNACK STAND', '13.00', '0.00', '0.73', '3.00', '00001',
'000014', '711', NULL);
```

Figure 7.3

The screenshot shows a SQL Server Management Studio window with four tabs at the top: SQLQuery7.sql - wp...UAD\jecasil1 (100)\*, SQLQuery6.sql - wp...UAD\jecasil1 (109)\*, SQLQuery5.sql - wp...SUAD\jecasil1 (81)\*, and SQLQuery4.sql - w. The fourth tab is active, displaying the following SQL code:

```
INSERT INVOICE VALUES ('200047', 'SNACK STAND', '13.00', '0.00', '0.73', '3.00', '00001',
'000014', '711', NULL)
```

Below the tabs, there is a large empty workspace. At the bottom of the window, the 'Messages' tab is selected, showing the following output:

```
100 % < 
Messages
(42 rows affected)
(1 row affected)
Completion time: 2022-04-23T21:11:19.0933031-07:00
```

New Invoice Insert Completed Successfully

Figure 7.4

SQLQuery7.sql - wp...UAD\jecasil1 (100)\* SQLQuery6.sql - wp...UAD\jecasil1 (109)\* SQLQuery5.sql - wp...SUAD\jecasil1 (81)\*

```
select *
from INVENTORY
```

100 %

Results Messages

	INVENTORY_ID	RESTOCK_LEVEL	ITEM_DESC	SERVING_SIZE	QOH	QOO
4	530	300	VEGGIE STICKS W/ DRESSING	NULL	459	0
5	531	300	VEGGIE STICKS W/ HUMMUS	NULL	-4	0
6	532	50	SIDE OF DRESSING	NULL	42	0
7	533	50	SIDE OF HUMMUS	NULL	31	100
8	550	300	SOFT PRETZEL BITES W/ CHEESE	NULL	460	0
9	551	300	SOFT PREZTEL BITES W/ MUSTARD	NULL	397	0
10	552	50	SIDE OF CHEESE	NULL	22	0
11	553	50	SIDE OF MUSTARD	NULL	11	100
12	580	300	SIDE OF FIRES	NULL	387	0
13	581	200	SIDE OF ONION RINGS	NULL	99	200
14	582	100	SIDE SALAD	NULL	129	0
15	590	500	HAMBURGER W/ CHEESE	NULL	99	0
16	595	250	VEGGIE BURGER W/ CHEESE	NULL	247	0
17	600	500	M&MS	NULL	41	0
18	601	500	PEANUT M&MS	NULL	232	100
19	610	300	SKITTLES	NULL	460	0
20	620	300	REDVINES	NULL	-2	0
21	630	200	JUNIOR MINTS	NULL	392	1000
22	640	100	GOOD & PLENTY	NULL	39	100
23	700	100	CHEESE PIZZA	8 IN	59	150
24	701	50	CHEESE PIZZA	10 IN	159	0
25	702	100	CHEESE PIZZA	14 IN	20	200
26	710	100	VEGGIE PIZZA	8 IN	5	150
27	711	50	VEGGIE PIZZA	10 IN	59	0
28	712	100	VEGGIE PIZZA	14 IN	-1	200
29	720	100	PEPPERONI PIZZA	8 IN	3	250
30	721	50	PEPPERONI PIZZA	10 IN	-4	100
31	722	100	PEPPERONI PIZZA	14 IN	119	0

Query executed successfully.

wpcmsql.cloud.w...

Inventory Table Reflecting REFINV Trigger Successfully Adjusted After Insert of New Invoice

Scripts for Loyalty Reward Level and Number of Visits Adjustments when a New Ticket is added:

```

create trigger LoyaltyLevel
on CUSTOMER
after update as
update CUSTOMER
set RWD_LVL_ID =
case
when NUM_VISIT < 5 then 'N/A'
when NUM_VISIT >= 1 and NUM_VISIT <5 then 'Emmy'
when NUM_VISIT >= 5 and NUM_VISIT <15 then 'Grammy'
when NUM_VISIT >= 15 and NUM_VISIT <30 then 'Oscar'
when NUM_VISIT >= 30 and NUM_VISIT <50 then 'Tony'
when NUM_VISIT >= 50 then 'EGOT'
end

```

```

CREATE TRIGGER LoyaltyLevel
ON CUSTOMER
AFTER Update AS
Update CUSTOMER
set RWD_LVL_ID =
case
WHEN NUM_VISIT < 5 THEN 'N/A'
when NUM_VISIT >= 1 AND NUM_VISIT <5 then 'Emmy'
when NUM_VISIT >= 5 AND NUM_VISIT <15 then 'Grammy'
when NUM_VISIT >= 15 AND NUM_VISIT <30 then 'Oscar'
when NUM_VISIT >= 30 AND NUM_VISIT <50 then 'Tony'
when NUM VISIT >= 50 THEN 'EGOT'

```

Messages  
Commands completed successfully.

Query executed successfully. | wpcmssql.cloud.wpcarey.asu.... | ASUAD\ejmoss1 (78) | BonlineB360 | 00:00:00 | 0 rows

Figure 7.5

### Successful Creation of Loyalty Reward Level Change Trigger

```
create trigger VISITS
on TICKET
after insert as
update CUSTOMER
set NUM_VISIT = NUM_VISIT + 1
where CUSTOMER_ID = (select CUSTOMER_ID from TICKET)
```

Figure 7.6

The screenshot shows the SQL Server Management Studio interface. The title bar displays two tabs: 'SQLQuery3.sql - w...SUAD\cebeck3 (88)\*' and 'SQLQuery2.sql - w...UAD\cebeck3'. The main window contains the T-SQL code for creating a trigger:

```
CREATE Trigger VISITS
ON Ticket
AFTER INSERT AS
UPDATE CUSTOMER
SET NUM_VISIT = NUM_VISIT + 1
WHERE CUSTOMER_ID = (SELECT CUSTOMER_ID FROM TICKET)
```

Below the code, the 'Messages' pane shows the output:

```
Commands completed successfully.
```

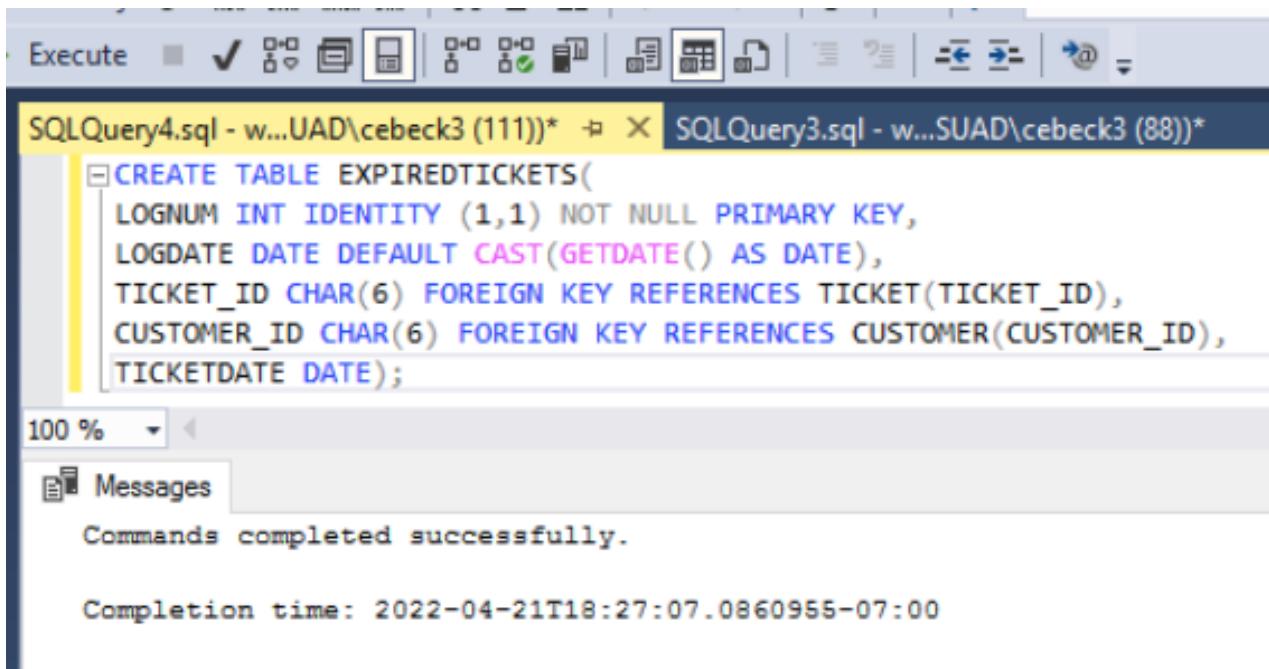
Completion time: 2022-04-21T18:24:49.6321845-07:00

Successful Creation of Adjustment to Number of Visits Trigger

Table, View, Trigger, and Procedure Scripts for Annual Customer Loyalty Level Update:Table Scripts:

```
create table EXPIREDTICKETS(  
LOGNUM int identity (1,1) NOT NULL PRIMARY KEY,  
LOGDATE date default cast (GETDATE() AS DATE),  
TICKET_ID int foreign key references TICKET(TICKET_ID),  
CUSTOMER_ID int foreign key references CUSTOMER(CUSTOMER_ID),  
TICKETDATE date);
```

Figure 7.7



The screenshot shows the SQL Server Management Studio interface. The title bar displays two tabs: "SQLQuery4.sql - w...UAD\cebeck3 (111)\*" and "SQLQuery3.sql - w...SUAD\cebeck3 (88)\*". The main window contains the T-SQL script for creating the EXPIREDTICKETS table. The script defines the table structure with columns LOGNUM, LOGDATE, TICKET\_ID, CUSTOMER\_ID, and TICKETDATE. The LOGNUM column is defined as an INT identity column with a value of 1,1 and is set as the primary key. The LOGDATE column is defined as a DATE type with a default value of GETDATE(). The TICKET\_ID and CUSTOMER\_ID columns are defined as INT types and are foreign keys referencing the TICKET and CUSTOMER tables respectively. The TICKETDATE column is defined as a DATE type. The status bar at the bottom shows the completion time as 2022-04-21T18:27:07.0860955-07:00.

```
CREATE TABLE EXPIREDTICKETS(  
LOGNUM INT IDENTITY (1,1) NOT NULL PRIMARY KEY,  
LOGDATE DATE DEFAULT CAST(GETDATE() AS DATE),  
TICKET_ID CHAR(6) FOREIGN KEY REFERENCES TICKET(TICKET_ID),  
CUSTOMER_ID CHAR(6) FOREIGN KEY REFERENCES CUSTOMER(CUSTOMER_ID),  
TICKETDATE DATE);
```

100 %

Messages

Commands completed successfully.

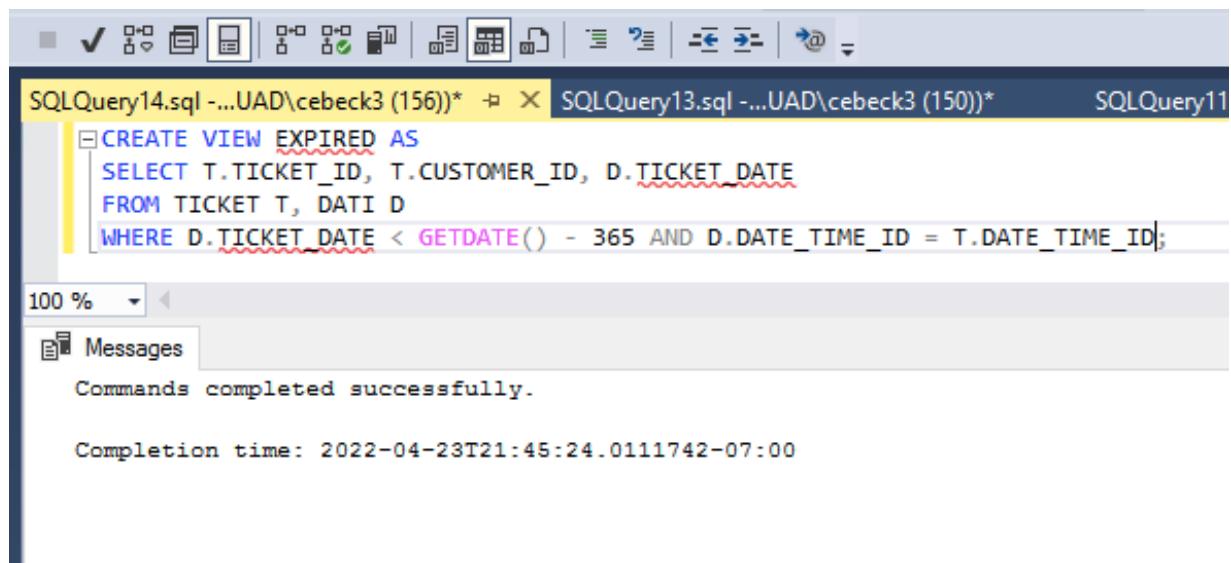
Completion time: 2022-04-21T18:27:07.0860955-07:00

Successful Creation of EXPIREDTICKETS Table

View Script:

```
create view EXPIRED as  
select T.TICKET_ID, T.CUSTOMER_ID, D.TICKET_DATE  
from TICKET T, DATI D  
where D.TICKET_DATE < GETDATE() - 365 and D.DATE_TIME_ID = T.DATE_TIME_ID;
```

Figure 7.8



The screenshot shows the SQL Server Management Studio interface. In the center pane, a query window displays the T-SQL code for creating a view named 'EXPIRED'. The code includes a SELECT statement with columns from tables 'TICKET' and 'DATI', filtered by a date condition. Below the code, the 'Messages' tab shows the output: 'Commands completed successfully.' and the completion time 'Completion time: 2022-04-23T21:45:24.0111742-07:00'. The top bar shows multiple tabs, and the title bar indicates the session is for 'SQLQuery11'.

```
CREATE VIEW EXPIRED AS  
SELECT T.TICKET_ID, T.CUSTOMER_ID, D.TICKET_DATE  
FROM TICKET T, DATI D  
WHERE D.TICKET_DATE < GETDATE() - 365 AND D.DATE_TIME_ID = T.DATE_TIME_ID;
```

Successful Creation of EXPIREDTICKETS View

Trigger Script:

```
create trigger NewVisits
on EXPIREDTICKETS
after insert as
update CUSTOMER
set NUM_VISIT = NUM_VISIT -
(select count(TICKET_ID) from CUSTOMER C, TICKET T
where T.CUSTOMER_ID = C.CUSTOMER_ID);
```

Figure 7.9

The screenshot shows a SQL Server Management Studio window. The title bar indicates two connections: 'SQLQuery10.sql - ...SUAD\cebeck3 (68)\*' and 'SQLQuery9.sql - w...SUAD\cebeck3 (66)'. The main pane displays the T-SQL script for creating the 'NewVisits' trigger. The script is as follows:

```
CREATE TRIGGER NewVisits
ON EXPIREDTICKETS
AFTER INSERT AS
UPDATE CUSTOMER
SET NUM_VISIT = NUM_VISIT -
(SELECT COUNT(TICKET_ID) FROM CUSTOMER C, TICKET T WHERE
T.CUSTOMER_ID = C.CUSTOMER_ID);
```

The 'Messages' pane at the bottom shows the output: 'Commands completed successfully.' and the completion time: 'Completion time: 2022-04-22T17:01:37.0522718-07:00'.

Successful Creation of NewVisits Trigger

Procedure Script:

```
create procedure LOYALTY as  
insert into EXPIREDTICKETS (TICKET_ID, CUSTOMER_ID, TICKETDATE)  
select TICKET_ID, CUSTOMER_ID, TICKET_DATE  
from EXPIRED  
where TICKET_ID not in (select TICKET_ID from EXPIREDTICKETS);
```

Figure 7.10

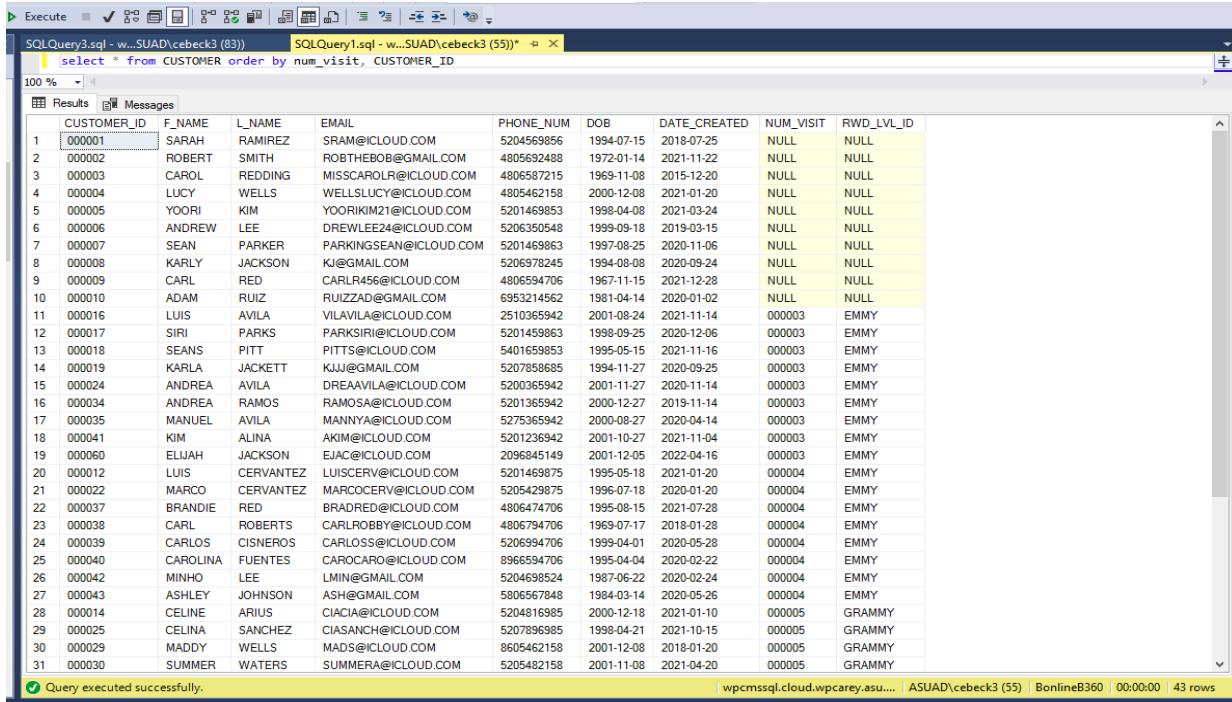
The screenshot shows the SQL Server Management Studio interface. The top bar has tabs for 'SQLQuery14.sql' and 'SQLQuery11.sql'. The main pane displays the T-SQL script for creating the 'LOYALTY' procedure. The script includes an 'INSERT INTO' statement that references the 'EXPIREDTICKETS' table, which is underlined in red, indicating it is a missing or unresolved object. The 'Messages' pane at the bottom shows the output: 'Commands completed successfully.' and the completion time: 'Completion time: 2022-04-23T21:47:58.3801509-07:00'.

```
CREATE PROCEDURE LOYALTY AS  
    INSERT INTO EXPIREDTICKETS(TICKET_ID, CUSTOMER_ID, TICKETDATE)  
        SELECT TICKET_ID, CUSTOMER_ID, TICKET_DATE  
        FROM EXPIRED  
    WHERE TICKET_ID NOT IN (SELECT TICKET_ID FROM  
        EXPIREDTICKETS);
```

Successful Creation of LOYALTY Procedure

Test Scripts for Loyalty Reward Level and Number of Visits Adjustments when a New Ticket is added:

Figure 7.11

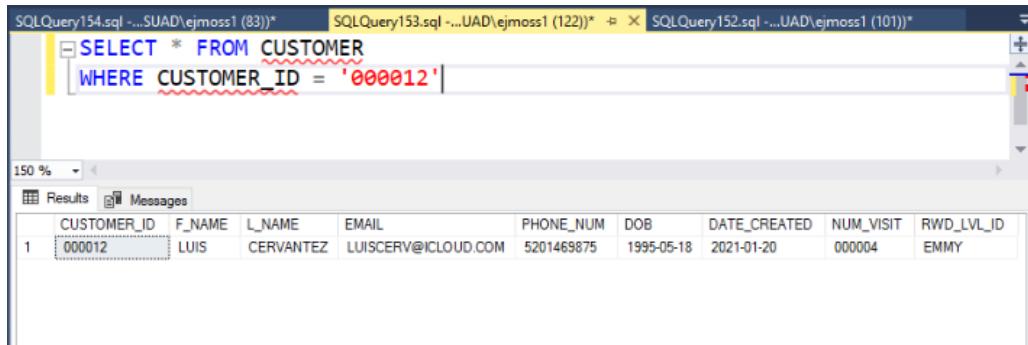


The screenshot shows a SQL Server Management Studio window with three tabs: 'Execute', 'Results', and 'Messages'. The 'Results' tab is selected and displays the output of a query:

```
SQLQuery1.sql - ...SUAD\cebeck3 (83) SQLQuery1.sql - ...UAD\ejmoss1 (122)* SQLQuery152.sql - ...UAD\ejmoss1 (101)*
select * from CUSTOMER order by num_visit, CUSTOMER_ID
```

The results show 43 rows of customer data. The columns are: CUSTOMER\_ID, F\_NAME, L\_NAME, EMAIL, PHONE\_NUM, DOB, DATE\_CREATED, NUM\_VISIT, and RWD\_LVL\_ID. The data includes various names like Sarah Ramirez, Robert Smith, Lucy Wells, etc., with their respective contact information and loyalty levels.

Customer Table before testing Loyalty Reward Level and Number of Visits Triggers and Procedures



The screenshot shows a SQL Server Management Studio window with three tabs: 'Execute', 'Results', and 'Messages'. The 'Results' tab is selected and displays the output of a query:

```
SQLQuery154.sql - ...SUAD\ejmoss1 (83)* SQLQuery153.sql - ...UAD\ejmoss1 (122)* SQLQuery152.sql - ...UAD\ejmoss1 (101)*
SELECT * FROM CUSTOMER
WHERE CUSTOMER_ID = '000012'
```

The results show one row of customer data for CUSTOMER\_ID 000012, named Luis Cervantez, with email LUISCERV@ICLOUD.COM and loyalty level EMMY.

Figure 7.12

A specific customer before creating new Ticket to test LoyaltyLevel Trigger

## New Ticket Test Script:

```
insert into TICKET values ('100047', 'A10018', '000012', '20120404113223', '900028');
```

Figure 7.13

```
SQLQuery9.sql - w...UAD\ejmoss1 (107)* SQLQuery1.sql - w...UAD\ejmoss1 (119)* SQLQuery3.sql - w...SUAD\ejmoss1 (89)* + ×
[ INSERT INTO TICKET VALUES ('100047', 'A10018', '000012', '20120404113223', '900028') ]
```

150 %

Messages

```
(56 rows affected)
(1 row affected)
(1 row affected)
```

150 %

Query executed successfully.

## New Test Ticket Insert Successfully Completed

Figure 7.14

Customer Table after testing Loyalty Reward Level and Number of Visits Triggers and Procedures

	CUSTOMER_ID	F_NAME	L_NAME	EMAIL	PHONE_NUM	DOB	DATE_CREATED	NUM_VISIT	RWD_LVL_ID
1	000001	SARAH	RAMIREZ	SRAM@ICLOUD.COM	5204569856	1994-07-15	2018-07-25	000001	N/A
2	000002	ROBERT	SMITH	ROBTHEBOB@GMAIL.COM	4805692483	1972-01-14	2021-11-22	000002	N/A
3	000003	CAROL	REDDING	MISSCAROLR@ICLOUD.COM	4806587215	1969-11-08	2015-12-20	000006	Grammy
4	000004	LUCY	WELLS	WELLSLUCY@ICLOUD.COM	4805462158	2000-12-08	2021-01-20	000017	Oscar
5	000005	YOORI	KIM	YOORIKIM21@ICLOUD.COM	5201469853	1998-04-08	2021-03-24	000005	Grammy
6	000006	ANDREW	LEE	DREWLEE24@ICLOUD.COM	5206350548	1999-09-18	2019-03-15	000031	Tony
7	000007	SEAN	PARKER	PARKINGSEAN@ICLOUD.COM	5201469863	1997-08-25	2020-11-06	000016	Oscar
8	000008	KARLY	JACKSON	KJ@GMAIL.COM	5206978245	1994-08-08	2020-09-24	000001	N/A
9	000009	CARL	RED	CARLR46@ICLOUD.COM	4806594706	1967-11-15	2021-12-28	000003	N/A
10	000010	ADAM	RUIZ	RUIZAD@GMAIL.COM	6953214562	1981-04-14	2020-01-02	000015	Oscar
11	000012	LUIS	CERVANTEZ	LUISCERV@ICLOUD.COM	5201469875	1995-05-18	2021-01-20	5	Grammy
12	000013	MAX	KITCHEN	KITCHENMAX@GMAIL.COM	5623148963	1999-11-14	2019-10-02	000015	Oscar
13	000014	CELINE	ARIUS	CIACIA@ICLOUD.COM	5204816985	2000-12-18	2021-01-10	000005	Grammy
14	000015	ANDRES	LOPEZ	DRESDRES@ICLOUD.COM	5204163259	2001-09-18	2020-04-19	000006	Grammy
15	000016	LUIS	AVILA	VILAVILA@ICLOUD.COM	2510365942	2001-08-24	2021-11-14	000003	N/A
16	000017	SIRI	PARKS	PARKSIRI@ICLOUD.COM	5201459863	1998-09-25	2020-12-06	000003	N/A
17	000018	SEANS	PITT	PITTS@ICLOUD.COM	5401659853	1995-05-15	2021-11-16	000003	N/A
18	000019	KARLA	JACKETT	KJJ@GMAIL.COM	5207856865	1994-11-27	2020-09-25	000003	N/A
19	000020	DANIELA	RUIZ	RUIZDANIELA@GMAIL.COM	5203214562	1979-06-21	2021-01-02	000015	Oscar
20	000021	DANNY	RUIZ	SICKODAN@GMAIL.COM	5204689523	2001-04-01	2020-05-02	000015	Oscar
21	000022	MARCO	CERVANTEZ	MARCOCERV@ICLOUD.COM	5205429875	1996-07-18	2020-01-20	000004	N/A
22	000023	ANDREW	GARCIA	DREWBOO@ICLOUD.COM	5204538548	1985-09-25	2020-04-17	000006	Grammy
23	000024	ANDREA	AVILA	DREAAVILA@ICLOUD.COM	5200365942	2001-11-27	2020-11-14	000003	N/A
24	000025	CELINA	SANCHEZ	CIASANCH@ICLOUD.COM	5207896985	1998-04-21	2021-10-15	000005	Grammy
25	000026	VICTOR	LOPEZ	VICYVIC@ICLOUD.COM	5203333259	2002-09-20	2021-04-19	000006	Grammy
26	000027	ADAMINA	LOPEZ	LOVEMINA@ICLOUD.COM	5203163456	2001-10-27	2021-04-24	000006	Grammy
27	000028	LOLA	LOPEZ	LOLOLA@ICLOUD.COM	4804163259	2000-12-18	2021-07-19	000006	Grammy
28	000029	MADDY	WELLS	MADS@ICLOUD.COM	8605462158	2001-12-08	2018-01-20	000005	Grammy
29	000030	SUMMER	WATERS	SUMMER@ICLOUD.COM	5205482158	2001-11-08	2021-04-20	000005	Grammy
30	000031	BRANDON	SMITH	BRANDONBOO@GMAIL.COM	4805682488	1976-01-17	2017-11-22	000015	Oscar

Figure 7.15

The same specific Customer Reflecting LoyaltyLevel and VISITS Trigger Successfully Adjusted After Insert of new Tickets

	CUSTOMER_ID	F_NAME	L_NAME	EMAIL	PHONE_NUM	DOB	DATE_CREATED	NUM_VISIT	RWD_LVL_ID
1	000012	LUIS	CERVANTEZ	LUISCERV@ICLOUD.COM	5201469875	1995-05-18	2021-01-20	5	Grammy

# TRANSACTIONS

## (SCRIPTS & SCREEN CAPTURE)

Figure 8.1

The screenshot shows a SQL Server Management Studio interface. At the top, there are three tabs: 'SQLQuery2.sql - wp...UAD\cigarc4 (88)\*', 'SQLQuery4.sql - wp...AD\cigarc4 (111)\*', and 'SQLQuery3.sql - wp...UAD\cigarc4 (79)\*'. The middle tab is active. Below the tabs, a query window contains the following SQL code:

```
Select *  
FROM invoice;
```

Below the query window is a results grid titled 'Results'. The grid displays data from the 'invoice' table. The columns are: INVOICE\_ID, DEPARTMENT, TOTAL, DISCOUNT, TAX, TIP, EMPLOYED\_ID, MENU\_ID, INVENTORY\_ID, and TICKET\_ID. The data is as follows:

	INVOICE_ID	DEPARTMENT	TOTAL	DISCOUNT	TAX	TIP	EMPLOYED_ID	MENU_ID	INVENTORY_ID	TICKET_ID
35	200036	BAR	2.00	0.00	0.12	0.00	000040	600036	400	NULL
36	200037	BAR	5.00	0.00	0.30	1.00	000031	600037	410	NULL
37	200038	BAR	10.00	0.00	0.60	2.00	000032	600038	420	NULL
38	200039	SNACK STAND	7.00	0.00	0.42	1.40	000041	600039	580	NULL
39	200040	SNACK STAND	7.00	0.00	0.42	1.40	000041	600040	581	NULL
40	200041	SNACK STAND	2.00	0.00	0.12	0.00	000041	600041	582	NULL
41	200042	SNACK STAND	2.00	0.00	0.12	0.00	000041	600042	590	NULL
42	200043	SNACK STAND	2.00	0.00	0.12	0.00	000041	600043	595	NULL

Invoice Table before creating new “Invoice Transactions”

New Invoice Transaction Scripts:

```
begin transaction;  
insert into INVOICE values ('200044', 'SNACK STAND', '5.00', '0.00', '.30', '1.00', '000041', '600028',  
'550', NULL);  
insert into INVOICE values ('200045', 'SNACK STAND', '7.00', '0.00', '.40', '2.00', '000041', '600019',  
'730', NULL);  
commit;
```

Figure 8.2

The screenshot shows the SQL Server Management Studio interface with three tabs open:

- SQLQuery2.sql - wp...UAD\cigarc4 (88)\*
- SQLQuery3.sql - wp...UAD\cigarc4 (79)\*
- SQLQuery1.sql - wp...UAD\cigarc4 (83)\*

The SQLQuery1.sql tab contains the following script:

```
BEGIN TRANSACTION  
insert into INVOICE values ('200044', 'SNACK STAND', '5.00', '0.00', '.30', '1.00', '000041', '600028', '550', NULL)  
insert into INVOICE values ('200045', 'SNACK STAND', '7.00', '0.00', '.40', '2.00', '000041', '600019', '730', NULL);  
COMMIT
```

The Messages tab shows the execution results:

```
(1 row affected)  
(1 row affected)  
Completion time: 2022-04-24T10:36:12.9861420+09:00
```

Invoice Transactions committed successfully

Figure 8.3

The screenshot shows three tabs at the top: SQLQuery2.sql, SQLQuery4.sql, and SQLQuery3.sql. The SQLQuery2.sql tab contains the query: "Select \* FROM invoice;". The results pane displays a table with 12 columns: INVOICE\_ID, DEPARTMENT, TOTAL, DISCOUNT, TAX, TIP, EMPLOYED\_ID, MENU\_ID, INVENTORY\_ID, and TICKET\_ID. There are 12 rows of data, each representing an invoice transaction.

	INVOICE_ID	DEPARTMENT	TOTAL	DISCOUNT	TAX	TIP	EMPLOYED_ID	MENU_ID	INVENTORY_ID	TICKET_ID
37	200038	BAR	10.00	0.00	0.60	2.00	000032	600038	420	NULL
38	200039	SNACK STAND	7.00	0.00	0.42	1.40	000041	600039	580	NULL
39	200040	SNACK STAND	7.00	0.00	0.42	1.40	000041	600040	581	NULL
40	200041	SNACK STAND	2.00	0.00	0.12	0.00	000041	600041	582	NULL
41	200042	SNACK STAND	2.00	0.00	0.12	0.00	000041	600042	590	NULL
42	200043	SNACK STAND	2.00	0.00	0.12	0.00	000041	600043	595	NULL
43	200044	SNACK STAND	5.00	0.00	0.30	1.00	000041	600028	550	NULL
44	200045	SNACK STAND	7.00	0.00	0.40	2.00	000041	600019	730	NULL

Invoice Table after successful commitment of Invoice Transactions

Figure 8.4

The screenshot shows three tabs at the top: SQLQuery3.sql, SQLQuery2.sql, and SQLQuery1.sql. The SQLQuery3.sql tab contains the query: "select \* from Ticket; select \* from MOVIE; select \* from customer; select \* from DATI;". The results pane displays four tables: Ticket, MOVIE, customer, and DATI. The Ticket table has columns: ID, I\_DATE, I\_TYPE, I\_LOCATION, I\_TIME, and I\_EMPLOYEE\_ID. The MOVIE table has columns: MOVIE\_ID, TITLE, RATING, and RELEASE\_YR. The customer table has columns: CUSTOMER\_ID, F\_NAME, L\_NAME, EMAIL, PHONE\_NUM, DOB, DATE\_CREATED, NUM\_VISIT, and RWD\_LVL\_ID. The DATI table has columns: DATE\_TIME\_ID, SCREEN\_ID, and MOVIE\_ID.

	I_DATE	I_TYPE	I_LOCATION	I_TIME	I_EMPLOYEE_ID
37	20121019101403	UUUUUZ	A10001	20121019101403	900001
38	100038	A10038	000038	20110504040550	900038
39	100039	A10039	000039	20120810095454	900039
40	100040	A10040	000040	20110505125500	900040

	MOVIE_ID	TITLE	RATING	RELEASE_YR
38	A10038	INGLORIOUS BASTERDS	R	2009-01-01
39	A10039	SLEEPING BEAUTY	G	1959-01-01
40	A10040	SNOW WHITE & THE SEVEN DWARVES	G	1937-01-01
41	A10041	UP	PG	2009-01-01

	CUSTOMER_ID	F_NAME	L_NAME	EMAIL	PHONE_NUM	DOB	DATE_CREATED	NUM_VISIT	RWD_LVL_ID
36	000037	BRAN...	RED	BRADRED@ICLOUD.COM	4906474706	1995-08-15	2021-07-28	000004	EMMY
37	000038	CARL...	ROBER...	CARLROBBY@ICLOUD.C...	4806794706	1969-07-17	2018-01-28	000004	EMMY
38	000039	CARLOS...	CISNER...	CARLOSG@ICLOUD.COM	5206994706	1999-04-01	2020-05-28	000004	EMMY
39	000040	CAROL...	FUENTES...	CAROCARO@ICLOUD.COM	8966594706	1995-04-04	2020-02-22	000004	EMMY
40	000041	KIM...	ALINA	AKIM@ICLOUD.COM	5201236942	2001-10-27	2021-11-04	000003	EMMY
41	000042	MINHO	LEE	LMIN@GMAIL.COM	5204698524	1987-06-22	2020-02-24	000004	EMMY
42	000043	ASHLEY	JOHNS...	ASH@GMAIL.COM	5806567848	1984-03-14	2020-05-26	000004	EMMY
43	000060	ELIJAH	JACKSON	EJAC@ICLOUD.COM	2096845149	2001-12-05	2022-04-16	000003	EMMY

	DATE_TIME_ID	SCREEN_ID	MOVIE_ID
31	20121019101403	101	A10001
32	20121119081405	102	A10002
33	20121203124503	116	A10016
34	20121212121212	124	A10024
35	20121230121211	127	A10027
36	20130101010101	114	A10014
37	20130101032121	131	A10031
38	20131225073055	103	A10003

Ticket Table before creating new “Ticket Transactions”

New Ticket Transaction Scripts:

```

begin transaction;

insert into TICKET values ('100041', 'A10025', '000043', '2011111111111', '101');
insert into TICKET values ('100042', 'A10027', '000017', '20121230121211', '28');
insert into TICKET values ('100043', 'A10010', '000003', '20110606125505', '33');

commit;

```

Figure 8.5

```

SQLQuery3.sql - wp...UAD\cigarc4 (96)*  SQLQuery1.sql - wp...UAD\cigarc4 (86)*
begin transaction;
insert into TICKET values ('100041', 'A10025', '000043', '2011111111111', '101');
insert into TICKET values ('100042', 'A10027', '000017', '20121230121211', '28');
insert into TICKET values ('100043', 'A10010', '000003', '20110606125505', '33');

Commit ;

```

80 %

Messages

(1 row affected)  
(1 row affected)  
(1 row affected)

Completion time: 2022-04-24T03:01:57.4813167+00:00

Ticket Transactions committed successfully

Figure 8.6

```

SQLQuery3.sql - wp...UAD\cigarc4 (96)*
Select *
from ticket ;

```

	TICKET_ID	MOVIE_ID	CUSTOMER_ID	DATE_TIME_ID	SEAT
36	100036	A10036	000036	20120203044554	900036
37	100037	A10002	000002	20121019101403	900001
38	100038	A10038	000038	20110504040550	900038
39	100039	A10039	000039	20120810095454	900039
40	100040	A10040	000040	20110505125505	900040
41	100041	A10027	100102	20121230121211	900025
42	100042	A10027	100103	20121230121211	900026
43	100043	A10027	100104	20121230121211	900027

Ticket Table after successful commitment of Ticket Transactions

## ACCESS: FORMS AND REPORTS

Add a record in one of the tables in Access (take a screenshot). Switch to the SQL Server database and show the new record (take a screenshot). -Celeste

Figure 9.1

CUSTOMER_ID	F_NAME	L_NAME	EMAIL	PHONE_NUM	DOB	DATE_CREATED	NUM_VISIT	RWD_LVL_ID
000036	VALDEZ	SARAH	SRAVV@ICLOUD 4804569856	1996-07-25	2017-08-27	000009	GRAMMY	
000037	RED	BRANDIE	BRADRED@ICLOUD 4806474706	1995-08-15	2021-07-28	000004	EMMY	
000038	ROBERTS	CARL	CARLROBBY@ICLOUD 4806794706	1969-07-17	2018-01-28	000004	EMMY	
000039	CINSNEROS	CARLOS	CARLOSS@ICLOUD 5206994706	1999-04-01	2020-05-28	000004	EMMY	
000040	FUENTES	CAROLINA	CAROCARO@ICLOUD 5966594706	1995-04-04	2020-02-22	000004	EMMY	
000041	KIM	ALINA	AKIM@ICLOUD 5201236942	2001-10-27	2021-11-04	000003	EMMY	
000042	LEE	MINHO	LMIN@GMAIL.COM 5204698524	1987-06-22	2020-02-24	000004	EMMY	

Figure 9.2

```
SELECT*
FROM CUSTOMER
WHERE F_NAME= 'MINHO';
```

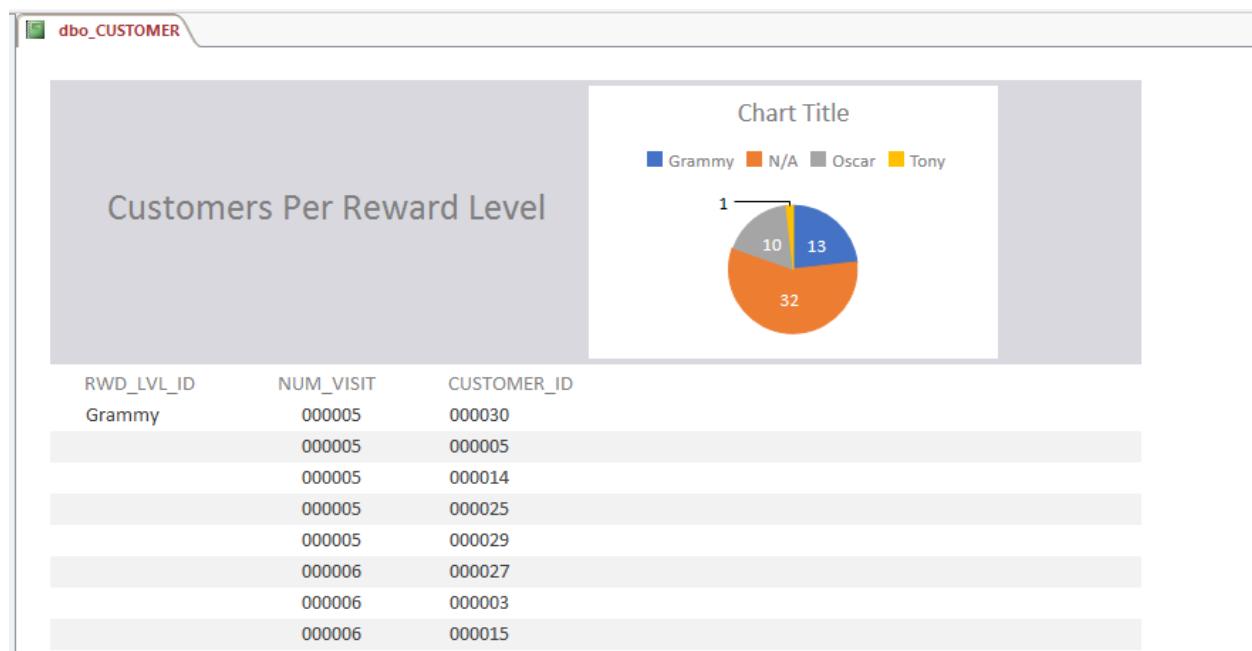
	CUSTOMER_ID	F_NAME	L_NAME	EMAIL	PHONE_NUM	DOB	DATE_CREATED	NUM_VISIT	RWD_LVL_ID
1	000042	MINHO	LEE	LMIN@GMAIL.COM	5204698524	1987-06-22	2020-02-24	00004	EMMY

Figure 9.3

CUSTOMER\_ID: 000001  
First Name: SARAH  
Last Name: RAMIREZ  
Email: SRAM@ICLOUD.COM  
Phone Number: 5204569856  
DOB: 1994-07-15  
Date Created: 2018-07-25  
Number of Visits: 000001  
Reward Level: EMMY

Form to add a new customer

Figure 9.4

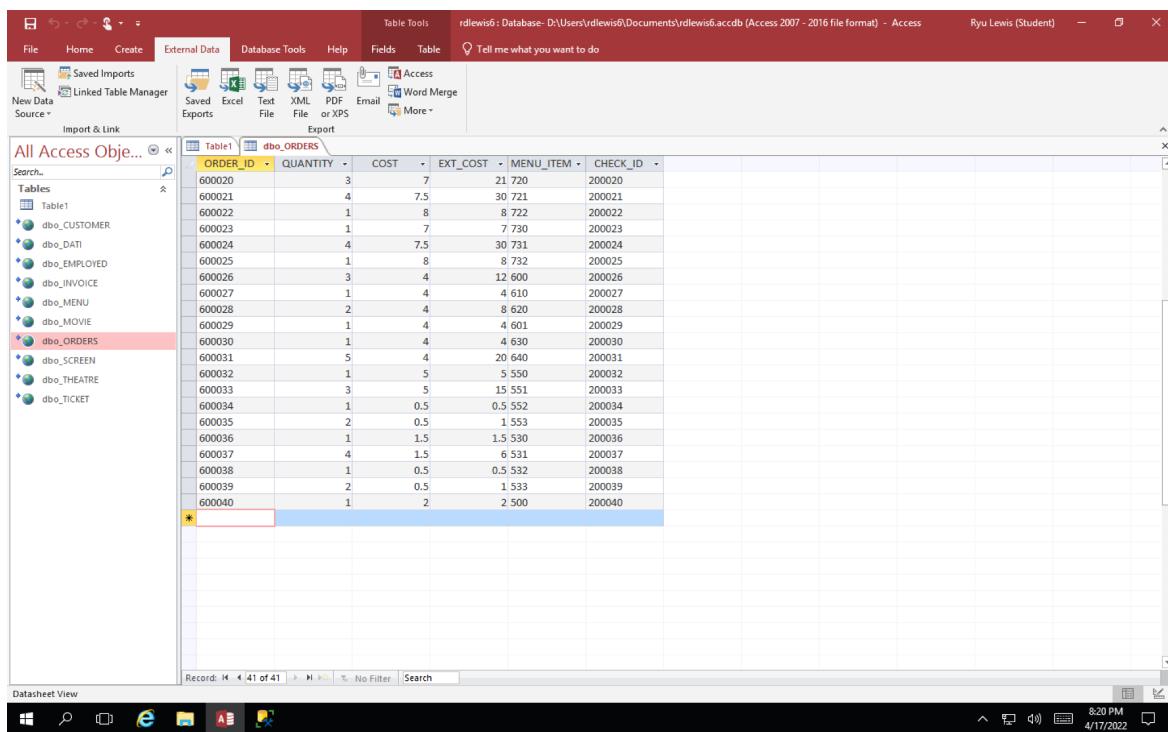


Report to show current number of Customers per Reward Level

Add a record in one of the tables in Access (take a screenshot). Switch to the SQL Server database and show the new record (take a screenshot). Ryu Lewis

Figure 9.5

Before



The screenshot shows the Microsoft Access application interface. The title bar indicates the database is 'rdlewis6.accdb'. The ribbon menu is visible with tabs like File, Home, Create, External Data, Database Tools, Help, Fields, and Table Tools. The 'External Data' tab is selected. The main area displays the 'dbo\_ORDERS' table in Datasheet View. The table contains 40 records with columns: ORDER\_ID, QUANTITY, COST, EXT\_COST, MENU\_ITEM, and CHECK\_ID. A new record is being added at the bottom of the table, indicated by a yellow asterisk (\*) in the first column. The status bar at the bottom shows 'Record: 41 of 41'.

ORDER_ID	QUANTITY	COST	EXT_COST	MENU_ITEM	CHECK_ID
600020	3	7	21 720	200020	
600021	4	7.5	30 721	200021	
600022	1	8	8 722	200022	
600023	1	7	7 730	200023	
600024	4	7.5	30 731	200024	
600025	1	8	8 732	200025	
600026	3	4	12 600	200026	
600027	1	4	4 610	200027	
600028	2	4	8 620	200028	
600029	1	4	4 601	200029	
600030	1	4	4 630	200030	
600031	5	4	20 640	200031	
600032	1	5	5 550	200032	
600033	3	5	15 551	200033	
600034	1	0.5	0.5 552	200034	
600035	2	0.5	1 553	200035	
600036	1	1.5	1.5 530	200036	
600037	4	1.5	6 531	200037	
600038	1	0.5	0.5 532	200038	
600039	2	0.5	1 533	200039	
600040	1	2	2 500	200040	

Figure 9.6

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays a tree view of database objects, including tables like CUSTOMER, EMPLOYEE, and ORDERS. In the center, a query window titled "SQLQuery2.sql - not connected" contains the SQL command: "select \* from orders". To the right of the command is a results grid showing the data from the ORDERS table. The results grid has columns: ORDER\_ID, QUANTITY, COST, EXT\_COST, MENU\_ITEM\_ID, and CHECK\_ID. The data consists of 40 rows of order details.

ORDER_ID	QUANTITY	COST	EXT_COST	MENU_ITEM_ID	CHECK_ID
27	600027	1	4.00	610	20027
28	600028	2	4.00	620	20028
29	600029	1	4.00	601	20029
30	600030	1	4.00	630	20030
31	600031	5	4.00	20.00	640
32	600032	1	5.00	5.00	550
33	600033	3	5.00	15.00	551
34	600034	1	0.50	0.50	552
35	600035	2	0.50	1.00	553
36	600036	1	1.50	1.50	530
37	600037	4	1.50	6.00	531
38	600038	1	0.50	0.50	532
39	600039	2	0.50	1.00	533
40	600040	1	2.00	2.00	500

Figure 9.7

After

The screenshot shows the Microsoft Access application interface. The title bar indicates the database is 'rdlewis6.accdb' located at 'D:\Users\rdlewis\Documents'. The ribbon menu is visible with tabs like File, Home, Create, External Data, Database Tools, Help, Fields, Table, and Tell me what you want to do. The 'External Data' tab is selected. The left pane shows the 'Tables' list with 'Table1' and several system tables like 'dbo\_CUSTOMER', 'dbo\_EMPLOYEE', etc. The main area displays the 'dbo\_ORDERS' table in Datasheet View. The table has six columns: ORDER\_ID, QUANTITY, COST, EXT\_COST, MENU\_ITEM, and CHECK\_ID. The data shows various order items with their respective quantities and costs. A specific row is highlighted with a yellow background and red border for ORDER\_ID 600041, which has a quantity of 1 and an extended cost of 5 510. The status bar at the bottom right shows the time as 8:22 PM and the date as 4/17/2022.

ORDER_ID	QUANTITY	COST	EXT_COST	MENU_ITEM	CHECK_ID
600020	3	7	21 720		200020
600021	4	7.5	30 721		200021
600022	1	8	8 722		200022
600023	1	7	7 730		200023
600024	4	7.5	30 731		200024
600025	1	8	8 732		200025
600026	3	4	12 600		200026
600027	1	4	4 610		200027
600028	2	4	8 620		200028
600029	1	4	4 601		200029
600030	1	4	4 630		200030
600031	5	4	20 640		200031
600032	1	5	5 550		200032
600033	3	5	15 551		200033
600034	1	0.5	0.5 552		200034
600035	2	0.5	1 553		200035
600036	1	1.5	1.5 530		200036
600037	4	1.5	6 531		200037
600038	1	0.5	0.5 532		200038
600039	2	0.5	1 533		200039
600040	1	2	2 500		200040
600041	1	5	5 510		200041

Figure 9.8

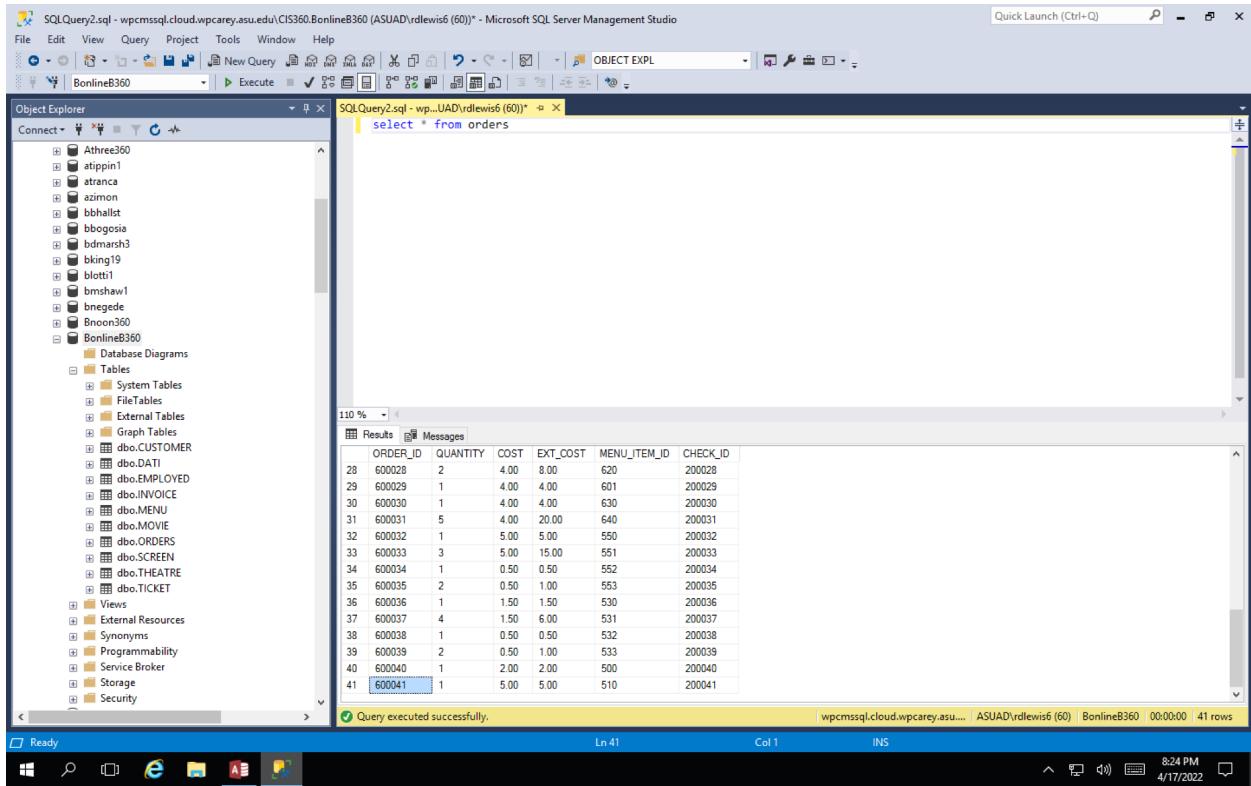
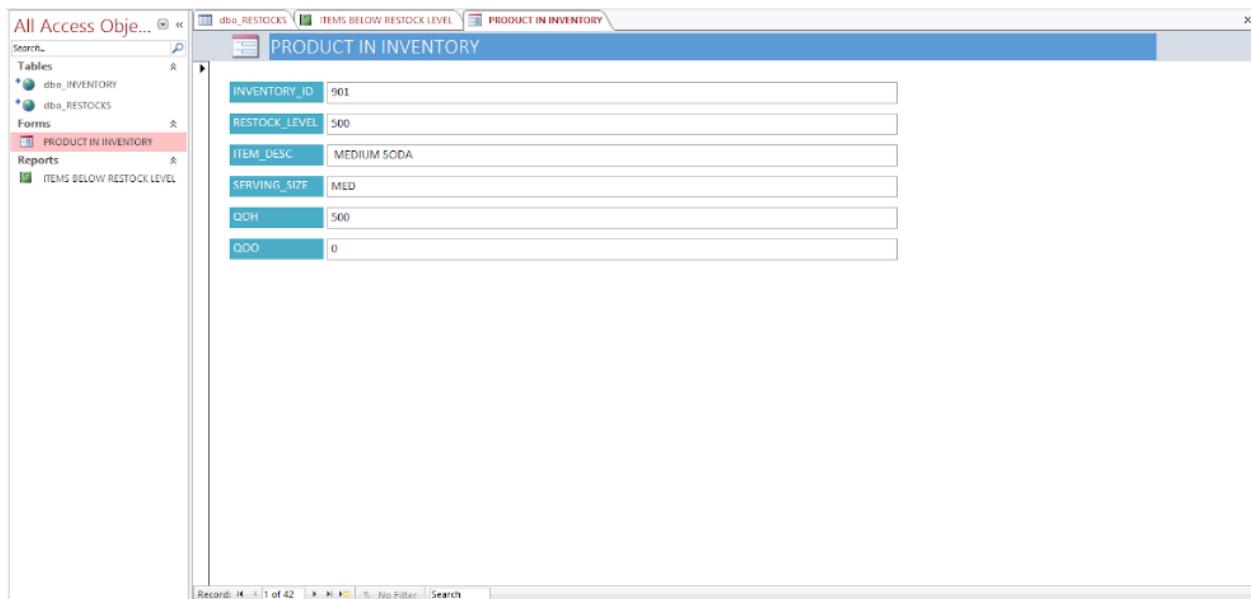


Figure 9.9



Form to add new inventory items to inventory

Figure 9.10

The screenshot shows a Microsoft Access application window. On the left, the 'All Access Obj... <<' navigation pane is visible, displaying 'Tables' (dbo\_INVENTORY, dbo\_RESTOCKS), 'Forms' (PRODUCT IN INVENTORY), and 'Reports' (ITEMS BELOW RESTOCK LEVEL). The main area is titled 'ITEMS BELOW RESTOCK LEVEL' and contains a table with the following data:

ITEM_DESC	QOH	RESTOCK_LEVEL
10 INCH CHEESE PIZZA	45	50
COCKTAIL	723	800
EXTRA LARGE POPCORN	686	700
LARGE POPCORN	688	700
M&MS	41	500
MEDIUM POPCORN	688	700
PEANUT M&MS	232	500
SIDE OF CHEESE	22	50
SIDE OF DRESSING	42	50
SIDE OF HUMMUS	31	50
SIDE OF MUSTARD	11	50
SMALL POPCORN	687	700
SMALL SODA	411	500
VEGGIE BURGER W/ CHEESE	247	250
VEGGIE STICKS W/ HUMMUS	250	300
WINE	789	800

At the bottom left, it says 'Sunday, April 24, 2022'. At the bottom right, it says 'Page 1 of 1'.

Report to show current restock levels/requirements

Add a new record in your SQL Server database (ALEC VALENCIA)

Figure 9.11

BEFORE:

The screenshot shows a SQL Server Management Studio interface. At the top, there is a title bar for 'SQLQuery1.sql - w...UAD\advalen4 (67)\*'. Below the title bar is a toolbar with several icons. The main area has two tabs: 'Results' (selected) and 'Messages'. In the 'Results' tab, there is a table of data. The table has 11 columns: CUSTOMER\_ID, F\_NAME, L\_NAME, EMAIL, PHONE\_NUM, DOB, DATE\_CREATED, NUM\_VISIT, RWD\_LVL\_ID, and two hidden columns. The data consists of 41 rows, each representing a customer entry. The columns correspond to the following data:

	CUSTOMER_ID	F_NAME	L_NAME	EMAIL	PHONE_NUM	DOB	DATE_CREATED	NUM_VISIT	RWD_LVL_ID
16	000017	SIRI	PARKS	PARKSIRI@ICLOUD.COM	5201459863	1998-09-25	2020-12-06	000003	EMMY
17	000018	SEANS	PITT	PITTS@ICLOUD.COM	5401659853	1995-05-15	2021-11-16	000003	EMMY
18	000019	KARLA	JACKETT	KJJJ@GMAIL.COM	5207858685	1994-11-27	2020-09-25	000003	EMMY
19	000020	DANIELA	RUIZ	RUIDANIELA@GMAIL.COM	5203214562	1979-06-21	2021-01-02	000015	OSCAR
20	000021	DANNY	RUIZ	SICKODAN@GMAIL.COM	5204689523	2001-04-01	2020-05-02	000015	OSCAR
21	000022	MARCO	CERVANTEZ	MARCOERCERV@ICLOUD.COM	5205429875	1996-07-18	2020-01-20	000004	EMMY
22	000023	ANDREW	GARCIA	DREWBBO@ICLOUD.COM	5204538548	1985-09-25	2020-04-17	000006	GRAMMY
23	000024	ANDREA	AVILA	DREAAVILA@ICLOUD.COM	5200365942	2001-11-27	2020-11-14	000003	EMMY
24	000025	CELINA	SANCHEZ	CIASANCHEZ@ICLOUD.COM	5207896985	1998-04-21	2021-10-15	000005	GRAMMY
25	000026	VICTOR	LOPEZ	VICYVIC@ICLOUD.COM	5203333259	2002-09-20	2021-04-19	000006	GRAMMY
26	000027	ADAMINA	LOPEZ	LOVEMINA@ICLOUD.COM	5203163456	2001-10-27	2021-04-24	000006	GRAMMY
27	000028	LOLA	LOPEZ	LOLOLA@ICLOUD.COM	4804163259	2000-12-18	2021-07-19	000006	GRAMMY
28	000029	MADDY	WELLS	MADS@ICLOUD.COM	8605462158	2001-12-08	2018-01-20	000005	GRAMMY
29	000030	SUMMER	WATERS	SUMMERA@ICLOUD.COM	5205482158	2001-11-08	2021-04-20	000005	GRAMMY
30	000031	BRAND...	SMITH	BRANDONBOO@GMAIL.COM	4805682488	1976-01-17	2017-11-22	000015	OSCAR
31	000032	ROBERT	SMART	ROBSMART@GMAIL.COM	4875692488	1978-01-14	2020-12-22	000015	OSCAR
32	000033	ZABDIEL	SANTOS	ZABI@GMAIL.COM	6205692488	1970-10-14	2021-10-02	000015	OSCAR
33	000034	ANDREA	RAMOS	RAMOSA@ICLOUD.COM	5201365942	2000-12-27	2019-11-14	000003	EMMY
34	000035	MANUEL	AVILA	MANNYA@ICLOUD.COM	5275365942	2000-08-27	2020-04-14	000003	EMMY
35	000036	SARAH	VALDEZ	SRAVV@ICLOUD.COM	4804569856	1996-07-25	2017-08-27	000009	GRAMMY
36	000037	BRANDIE	RED	BRADRED@ICLOUD.COM	4806474706	1995-08-15	2021-07-28	000004	EMMY
37	000038	CARL	ROBERTS	CARLROBBY@ICLOUD.COM	4806794706	1969-07-17	2018-01-28	000004	EMMY
38	000039	CARLOS	CISNEROS	CARLOSS@ICLOUD.COM	5206994706	1999-04-01	2020-05-28	000004	EMMY
39	000040	CAROLI...	FUENTES	CAROCARO@ICLOUD.COM	8966594706	1995-04-04	2020-02-22	000004	EMMY
40	000041	KIM	ALINA	AKIM@ICLOUD.COM	5201236942	2001-10-27	2021-11-04	000003	EMMY
41	000042	MINHO	LEE	LMIN@GMAIL.COM	5204698524	1987-06-22	2020-02-24	000004	EMMY

Figure 9.12

AFTER:

AFTER:

SQLQuery1.sql - w...UAD\advalen4 (67)\* ↗ X

```


INSERT INTO CUSTOMER
VALUES ('000043', 'ASHLEY', 'JOHNSON', 'ASH@GMAIL.COM', '5806567848', '1984-03-14', '2020-05-26', '000004', 'GRAMMY');
SELECT *
FROM CUSTOMER;


```

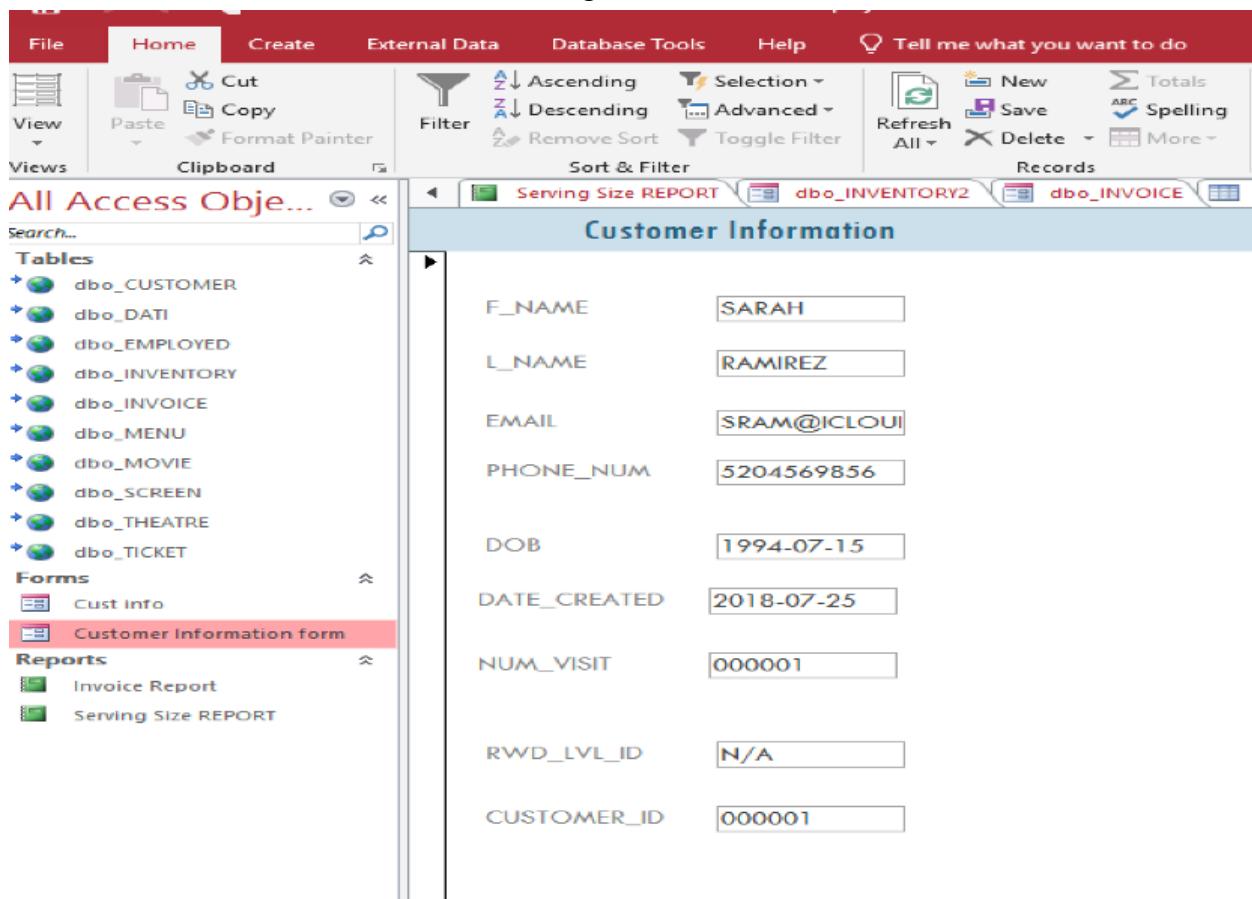
100 %

	CUSTOMER_ID	F_NAME	L_NAME	EMAIL	PHONE_NUM	DOB	DATE_CREATED	NUM_VISIT	RWD_LVL_ID
17	000018	SEANS	PITT	PITTS@ICLOUD.COM	5401659853	1995-05-15	2021-11-16	000003	EMMY
18	000019	KARLA	JACKETT	KJJ@GMAIL.COM	5207858685	1994-11-27	2020-09-25	000003	EMMY
19	000020	DANIELA	RUIZ	RUIZDANIELA@GMAIL.COM	5203214562	1979-06-21	2021-01-02	000015	OSCAR
20	000021	DANNY	RUIZ	SICKODAN@GMAIL.COM	5204689523	2001-04-01	2020-05-02	000015	OSCAR
21	000022	MARCO	CERVANTEZ	MARCOCERV@ICLOUD.COM	5205429875	1996-07-18	2020-01-20	000004	EMMY
22	000023	ANDREW	GARCIA	DREWBOO@ICLOUD.COM	5204538548	1985-09-25	2020-04-17	000006	GRAMMY
23	000024	ANDREA	AVILA	DREAAVILA@ICLOUD.COM	5200365942	2001-11-27	2020-11-14	000003	EMMY
24	000025	CELINA	SANCHEZ	CIASANCH@ICLOUD.COM	5207896985	1998-04-21	2021-10-15	000005	GRAMMY
25	000026	VICTOR	LOPEZ	VICYVIC@ICLOUD.COM	5203333259	2002-09-20	2021-04-19	000006	GRAMMY
26	000027	ADAMINA	LOPEZ	LOVEMINA@ICLOUD.COM	5203163456	2001-10-27	2021-04-24	000006	GRAMMY
27	000028	LOLA	LOPEZ	LOLOLA@ICLOUD.COM	4804163259	2000-12-18	2021-07-19	000006	GRAMMY
28	000029	MADDY	WELLS	MADS@ICLOUD.COM	8605462158	2001-12-08	2018-01-20	000005	GRAMMY
29	000030	SUMMER	WATERS	SUMMERA@ICLOUD.COM	5205492158	2001-11-08	2021-04-20	000005	GRAMMY
30	000031	BRAND...	SMITH	BRANDONBOO@GMAIL.COM	4805682488	1976-01-17	2017-11-22	000015	OSCAR
31	000032	ROBERT	SMART	ROBSMART@GMAIL.COM	4875692488	1978-01-14	2020-12-22	000015	OSCAR
32	000033	ZABDIEL	SANTOS	ZABI@GMAIL.COM	6205692488	1970-10-14	2021-10-02	000015	OSCAR
33	000034	ANDREA	RAMOS	RAMOSA@ICLOUD.COM	5201365942	2000-12-27	2019-11-14	000003	EMMY
34	000035	MANUEL	AVILA	MANNYA@ICLOUD.COM	5275365942	2000-08-27	2020-04-14	000003	EMMY
35	000036	SARAH	VALDEZ	SRAVV@ICLOUD.COM	4804569856	1996-07-25	2017-08-27	000009	GRAMMY
36	000037	BRANDIE	RED	BRADRED@ICLOUD.COM	4806474706	1995-08-15	2021-07-28	000004	EMMY
37	000038	CARL	ROBERTS	CARLROBBY@ICLOUD.COM	4806794706	1969-07-17	2018-01-28	000004	EMMY
38	000039	CARLOS	CISNEROS	CARLOSS@ICLOUD.COM	5206994706	1999-04-01	2020-05-28	000004	EMMY
39	000040	CAROLI...	FUENTES	CAROCARO@ICLOUD.COM	89666594706	1995-04-04	2020-02-22	000004	EMMY
40	000041	KIM	ALINA	AKIM@ICLOUD.COM	5201236942	2001-10-27	2021-11-04	000003	EMMY
41	000042	MINHO	LEE	LMIN@GMAIL.COM	5204698524	1987-06-22	2020-02-24	000004	EMMY
42	000043	ASHLEY	JOHNSON	ASH@GMAIL.COM	5806567848	1984-03-14	2020-05-26	000004	GRAMMY

Figure 9.13

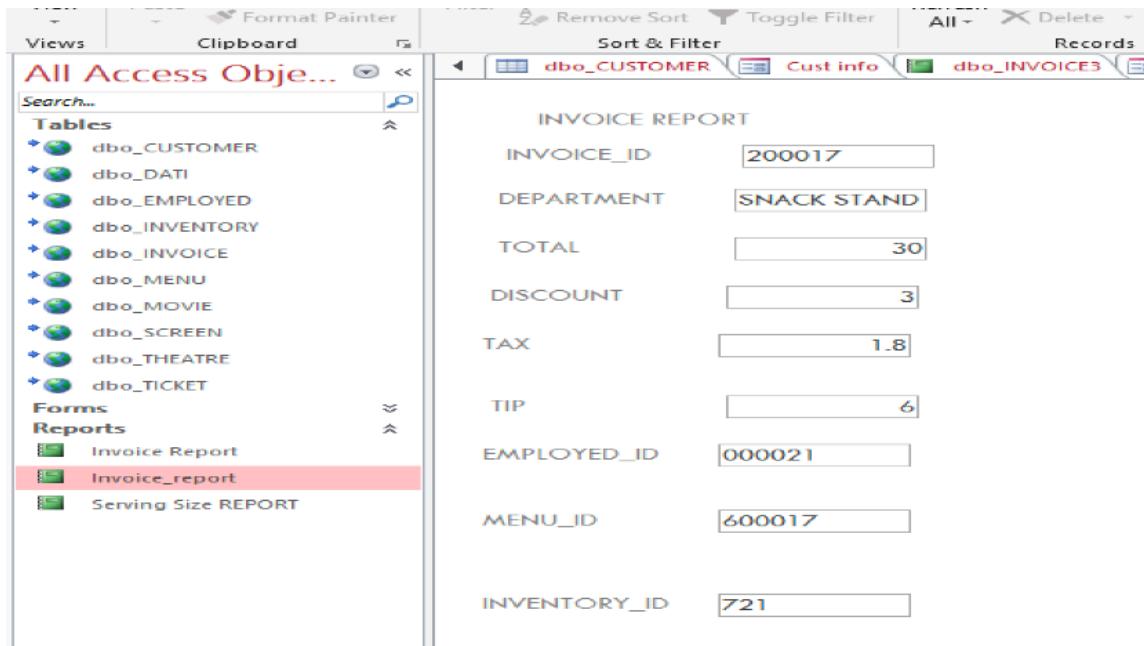
ACCESS: (ALEC VALENCIA)

Figure 9.14



An alternate form to add a new customer

Figure 9.15



Report showing current open invoice

Figure 9.16

Before: by Jorge Castro

SQLQuery90.sql - w...ASUAD\jecasit1 (108)*												
Select * From EMPLOYED;												
100 %												
EMPLOYED_ID	F_NAME	INITIAL	L_NAME	ADDRESS	CITY	STATE	ZIP	PHN_NUM	EMAIL	MANAGER_ID	theatre_id	
18 000018	LANE	B	DESPAIN	1399 W JUSTICE ST.	TEMPE	AZ	85004	6027859333	LANE.DESPAN@GMAIL.COM	000008	105	
19 000019	UMA	T	STRONG	1569 W STRAW ST.	TEMPE	AZ	85004	6028086112	UMA.STRONG@GMAIL.COM	000007	106	
20 000020	BILLY	W	MCOON	1776 E INDEPENDENC.	TEMPE	AZ	85005	6027083076	BILLY.MCDONALD@GMAIL.C...	000006	107	
21 000021	MANNY	P	BLUE	2011 E YOUNG ST.	TEMPE	AZ	85002	6028906134	MANNY.BLUE@GMAIL.COM	000005	108	
22 000022	ERIN	L	DESTINY	1592 S TRUE ST.	TEMPE	AZ	85002	6023851290	ERIN.DESTINY@GMAIL.COM	000004	109	
23 000023	POLLY	B	MCARTH	1224 E STRAWBERRY...	TEMPE	AZ	85001	4807162253	POLLY.MCARTHUR@GMAIL...	000003	110	
24 000024	BRANDON	J	BUNCH	3907 W TREE ST.	TEMPE	AZ	85005	6028904321	BRANDON.BUNCH@GMAIL...	000002	111	
25 000025	KATYA	S	BERRY	3400 S APPLETREE ST.	TEMPE	AZ	85004	6029011542	KATYA.BERRY@GMAIL.COM	000001	112	
26 000026	STEVEN	C	AUSTIN	2300 N HOUSTON ST.	TEMPE	AZ	85004	4806709327	STEVEN.AUSTIN@GMAIL.COM	000002	111	
27 000027	KATIE	R	CHRISTI...	2809 E COOKIE ST.	TEMPE	AZ	85002	6020997184	KATIE.CHRISTIANSEN@GMA...	000003	110	
28 000028	NINA	W	BALLARD	4003 S LIME ST.	TEMPE	AZ	85001	602021185	NINA.BALLARD@GMAIL.COM	000004	109	
29 000029	GREG	F	RIGGS	1875 E RURAL ST.	TEMPE	AZ	85006	6028471904	GREG.RIGGS@GMAIL.COM	000005	108	
30 000030	CRISTY	A	VICTOR	4560 S SILVER ST.	TEMPE	AZ	85002	4807264002	CHRISTY.VICTOR@GMAIL.C...	000006	107	
31 000031	STELLA	O	RICKS	3609 W BIRCH ST.	TEMPE	AZ	85005	4805070333	STELLA.RICKS@GMAIL.COM	000007	106	
32 000032	CARME	P	LOW	3001 E SPRUCE STAY ST.	TEMPE	AZ	85005	4805070335	CARME.LOW@GMAIL.COM	000008	105	
33 000033	BLAKE	R	GARRETT	400 S STONY ST.	TEMPE	AZ	85004	4805483772	BLAKE.GARRETT@GMAIL.COM	000009	104	
34 000034	ANNIE	E	OWENS	4700 W MOON ST.	TEMPE	AZ	85002	6023459095	ANNIE.OWENS@GMAIL.COM	000010	103	
35 000035	SUNNY	K	SPENCER	2740 E PILOT ST.	TEMPE	AZ	85005	4805382494	SUNNY.SPENCER@GMAIL.C...	000011	102	
36 000036	FRED	G	BUBBLES	2658 W TEA ST.	TEMPE	AZ	85003	4805045829	FRED.BUBBLES@GMAIL.COM	000012	101	
37 000037	HALEY	E	SANDERS	4508 S ROSE ST.	TEMPE	AZ	85005	4805903348	HALEY.SANDERS@GMAIL.C...	000012	101	
38 000038	HARRY	B	GRAHAM	100 S IGLOO ST.	TEMPE	AZ	85005	4805007182	HARRY.GRAHAM@GMAIL.COM	000012	101	
39 000039	CHARLIE	C	WALKER	7200 S BULLHORN ST.	TEMPE	AZ	85005	4801110972	CHARLIE.WALKER@GMAIL.C...	000011	102	
40 000040	MIA	B	FOX	3705 E CALICO ST.	TEMPE	AZ	85005	4808080492	MIA.FOX@GMAIL.COM	000010	103	

Query executed successfully.

Figure 9.17

After: by Jorge Castro

SQLQuery91.sql - w...SUAD\jecasil1 (85)\* → SQLQuery90.sql - w...UAD\jecasil1 (108)\*

```
Insert into Employee VALUES
('0000041', 'JACQUELINE', 'M', 'QWERTY', '4356 N WILLIAMS RD.', 'GILBERT', 'AZ', '85142', '4899874353', 'JQWERTY@EMAIL.COM', '000011', '109')
```

(1 row affected)

Completion time: 2022-04-17T16:30:40.2245226-07:00

100 %

Messages

Query executed successfully.

SQLQuery91.sql - w...SUAD\jecasil1 (85)\* SQLQuery90.sql - w...UAD\jecasil1 (108)\*

```
Select
From EMPLOYEE;
```

Results

EMPLOYEE_ID	F_NAME	INITIAL	L_NAME	ADDRESS	CITY	STATE	ZIP	PHONE_NUM	EMAIL	MANAGER_ID	theatre_id
29	GREG	F	RIGGS	1875 E RURAL ST.	TEMPE	AZ	85006	6220471904	GREG.RIGGS@GMAIL.COM	000005	108
30	CHRISTY	A	VICTOR	4560 S SILVER ST.	TEMPE	AZ	85002	4807254002	CHRISTY.VICTOR@GMAIL.COM	000006	107
31	STELLA	O	RICKS	3608 W YUMA ST.	TEMPE	AZ	85002	4800096333	STELLA.RICKS@GMAIL.COM	000007	106
32	CARRIE	P	LION	3900 E BROADWAY ST.	TEMPE	AZ	85005	480907766	CARRIE.LION@GMAIL.COM	000008	105
33	BLAKE	R	GARRETT	900 E SYDNEY ST.	TEMPE	AZ	85004	48094372	BLAKE.GARRETT@GMAIL.COM	000009	104
34	ANNIE	E	OWENS	4700 W MOON ST.	TEMPE	AZ	85002	6023459098	ANNIE.OWENS@GMAIL.COM	000010	103
35	SUNNY	K	SPENCER	2740 E PILOT ST.	TEMPE	AZ	85005	480932494	SUNNY.SPENCER@GMAIL.COM	000011	102
36	FRED	G	BUBBLES	2668 W TEA ST.	TEMPE	AZ	85003	4808045829	FRED.BUBBLES@GMAIL.COM	000012	101
37	HALEY	E	SANDERS	4508 S ROSE ST.	TEMPE	AZ	85005	48080930348	HALEY.SANDERS@GMAIL.COM	000012	101
38	HARRY	R	GRAHAM	100 S IGLOO ST.	TEMPE	AZ	85005	4808077182	HARRY.GRAHAM@GMAIL.COM	000012	101
39	CHARLIE	C	WALKER	7200 S BULLHORN ST.	TEMPE	AZ	85005	4801110972	CHARLIE.WALKER@GMAIL.COM	000011	102
40	MIA	B	FOX	3705 E CALICO ST.	TEMPE	AZ	85005	4800804902	MIA.FOX@GMAIL.COM	000010	103
41	JACQUELINE	M	QWERTY	4356 N WILLIAMS RD.	GILBERT	AZ	85142	4809874353	JQWERTY@EMAIL.COM	000011	109

Query executed successfully.

Figure 9.18

TITLE	RATING	MOVIE_ID	RELEASE_YR
HARRY POTTER & THE SORCERER PG	PG-13	A10001	2001-01-01
TITANIC	PG-13	A10002	1997-01-01
STAR WARS EPISODE IX	PG-13	A10003	2019-01-01
BACK TO THE FUTURE	PG	A10004	1985-01-01
JAWS	PG	A10005	1975-01-01
JURASSIC PARK	PG-13	A10006	1993-01-01
GHOST	PG-13	A10007	1990-01-01
LION KING	G	A10008	1994-01-01
TOY STORY	G	A10009	1995-01-01
THE AVENGERS	PG-13	A10010	2012-01-01
GIRLSE	PG	A10011	1978-01-01
ROCKY	PG	A10012	1976-01-01
HARRY POTTER & CHAMBER OF S	PG	A10013	2002-01-01
HARRY POTTER & PRISONER OF / PG		A10014	2004-01-01
HARRY POTTER & GOBLET OF FIR PG-13		A10015	2005-01-01
HARRY POTTER & ORDER OF THE PG-13		A10016	2007-01-01
HARRY POTTER & HALF BLOOD P PG		A10017	2009-01-01
HARRY POTTER & DEATHLY HALL PG-13		A10018	2010-01-01
HARRY POTTER & DEATHLY HALL PG-13		A10019	2011-01-01
STAR WARS	PG	A10020	1977-01-01
STAR WARS EPISODE V	PG	A10021	1980-01-01
STAR WARS EPISODE VI	PG-13	A10022	1983-01-01
STAR WARS EPISODE I	PG	A10023	1999-01-01
STAR WARS EPISODE II	PG	A10024	2002-01-01
STAR WARS EPISODE III	PG	A10025	2005-01-01
STAR WARS EPISODE VII	PG-13	A10026	2015-01-01
STAR WARS EPISODE VIII	PG-13	A10027	2017-01-01
THE AVENGERS: AGE OF ULTRON PG-13		A10028	2015-01-01
TOY STORY 2	G	A10029	1999-01-01
TOY STORY 3	G	A10030	2010-01-01
TOY STORY 4	G	A10031	2019-01-01
THE AVENGERS: INFINITY WAR PG-13		A10032	2018-01-01
THE AVENGERS: END GAME PG-13		A10033	2019-01-01
BACK TO THE FUTURE P2	PG	A10034	1989-01-01
BACK TO THE FUTURE P3	PG	A10035	1990-01-01
KILL BILL: VOL 1	R	A10036	2003-01-01
KILL BILL: VOL 2	R	A10037	2004-01-01
INGLORIOUS BASTERDS	R	A10038	2009-01-01
SLEEPING BEAUTY	G	A10039	1959-01-01
SNOW WHITE & THE SEVEN DW G	G	A10040	1937-01-01
UP	PG	A10041	2009-01-01
DESPICABLE ME	PG	A10042	2010-01-01

Add / Edit Movie

TITLE	<input type="text"/>
RATING	<input type="text"/>
MOVIE_ID	<input type="text"/>
RELEASE_YR	<input type="text"/>
<input type="button" value="Add Movie"/> <input type="button" value="Save Edit"/>	
<input type="button" value="Delete Movie"/>	

Form to add a new movie to the screening list

Figure 9.19

MOVIES			
MOVIE_ID	TITLE	RATING	RELEASE_YR
A10001	HARRY POTTER & THE SORCERER'S STONE	PG	2001-01-01
A10002	ETANIC	PG-13	1997-01-01
A10003	STAR WARS EPISODE IX	PG-13	2019-01-01
A10004	BAC TO THE FUTURE	PG	1985-01-01
A10005	JAWS	PG	1975-01-01
A10006	JURASSIC PARK	PG-13	1993-01-01
A10007	THE GODFATHER	PG-13	1972-01-01
A10008	LION KING	G	1994-01-01
A10009	TOY STORY	G	1995-01-01
A10010	THE AVENGERS	PG-13	2012-01-01
A10011	CRAZY	PG	1976-01-01
A10012	DEADLY	PG	1976-01-01
A10013	HARRY POTTER & CHAMBER OF SECRETS	PG	2002-01-01
A10014	HARRY POTTER & ORDER OF THE PHANTOM	PG	2004-01-01
A10015	HARRY POTTER & PRISONER OF AZKABAN	PG-13	2004-01-01
A10016	HARRY POTTER & ORDER OF THE PHANTOM	PG-13	2007-01-01
A10017	HARRY POTTER & HALF BLOOD PRINCE	PG	2009-01-01
A10018	HARRY POTTER & DEATHLY HALLOWS - PT	PG-13	2010-01-01

MOVIES			
MOVIE_ID	TITLE	RATING	RELEASE_YR
A10019	HARRY POTTER & THE HALF-BLOOD PRINCE	PG-13	2011-01-01
A10020	STAR WARS	PG	1977-01-01
A10021	STAR WARS EPISODE V	PG	1980-01-01
A10022	STAR WARS EPISODE VI	PG-13	1983-01-01
A10023	STAR WARS EPISODE VII	PG	1999-01-01
A10024	STAR WARS EPISODE II	PG	2002-01-01
A10025	STAR WARS EPISODE III	PG	2005-01-01
A10026	STAR WARS EPISODE IV	PG-12	2019-01-01
A10027	STAR WARS EPISODE V	PG-12	2017-01-01
A10028	THE AVENGERS: AGE OF ULTRON	PG-13	2015-01-01
A10029	TOY STORY 2	G	1999-01-01
A10030	TOY STORY 3	G	2010-01-01
A10031	TOY STORY 4	G	2019-01-01
A10032	THE AVENGERS: INFINITY WAR	PG-13	2018-01-01
A10033	THE AVENGERS: END GAME	PG-12	2019-01-01
A10034	STAR WARS: FUTURE PS	PG	1988-01-01
A10035	BACK TO THE FUTURE	PG	1990-01-01
A10036	KILL BILL: VOL. 1	R	2003-01-01
A10037	KILL BILL: VOL. 2	R	2004-01-01
A10038	THREE HOUR BATTERRY	R	2009-01-01
A10039	SLEEPING BEAUTY	G	1959-01-01

Report showing all movies showing at Palermo

Create a form for one of your tables or for a view in Access; your form must look like a form – not simply like an Access data table. Use the Building Access Forms tutorial located on Canvas. Take a snapshot of your form (with data).

Figure 9.20

EMPLOYEE FORM (ERIKA MOSS):

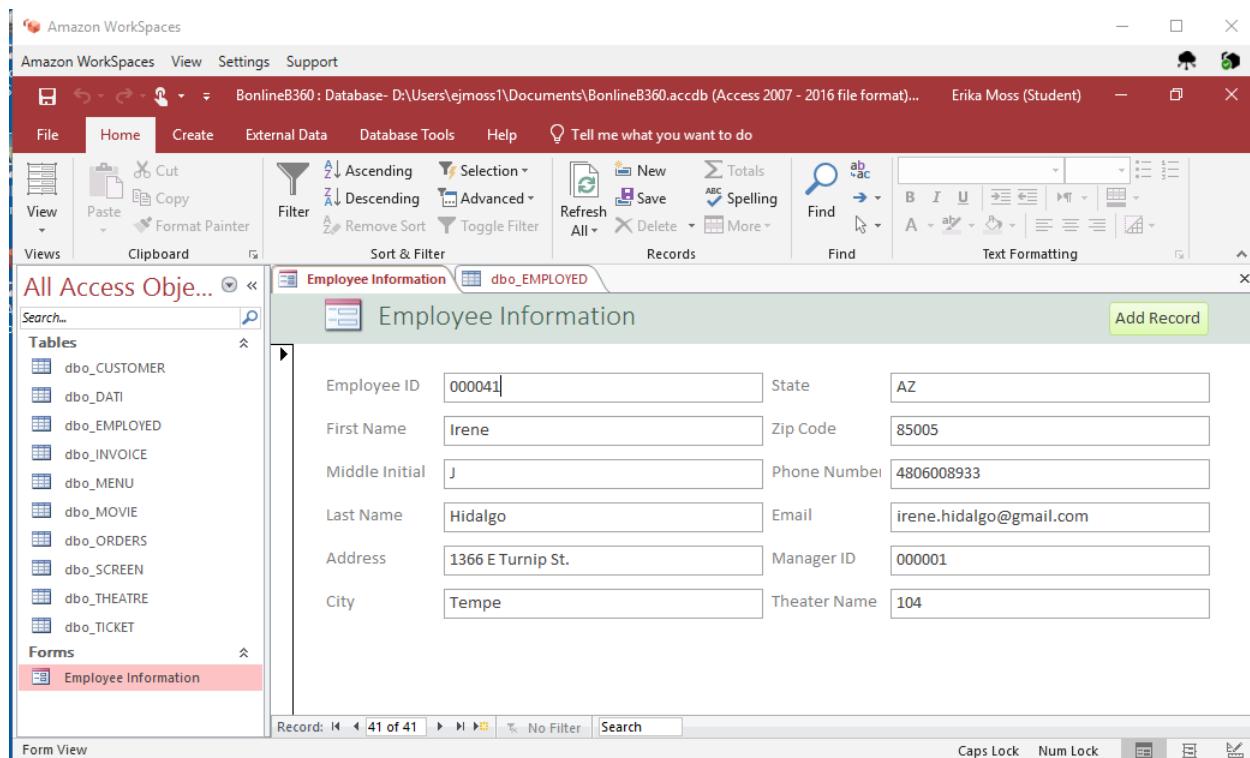


Figure 9.21

This screenshot shows the "Employee Information" form from a different perspective, likely a simplified view or a different version of the application. The form fields are identical to Figure 9.20: Employee ID (000041), First Name (Irene), Middle Initial (J), Last Name (Hidalgo), Address (1366 E Turnip St.), City (Tempe), State (AZ), Zip Code (85005), Phone Number (4806008933), Email (irene.hidalgo@gmail.com), Manager ID (000001), and Theater Name (104). An "Add Record" button is located in the top right corner. The form title is "Employee Information".

Form to add a new Employee

Figure 9.22

dbo_Jacqueline Qwerty	
Jacqueline Qwerty Invoices	
EMPLOYED_ID	INVOICE_ID
000041	200039
000041	200040
000041	200041
000041	200042
000041	200043
000041	200044
000041	200045

Sample Report to show Invoices per Employee

Figure 9.23  
CUSTOMER FORM (ELIJAH JACKSON):

The screenshot shows the frmCustomers form in Microsoft Access. The title bar says "frmCustomers". The main area is titled "Customer Information". It contains four text boxes: CUSTOMER\_ID (000060), EMAIL (EJAC@ICLOUD.COM), F\_NAME (ELIJAH), and PHONE\_NUM (2096845149). Below these are two more text boxes: L\_NAME (JACKSON) and RWD\_LVL\_ID (OSCAR). On the left, there's a navigation pane titled "All Access Objekte" with sections for "Tables" and "Forms". Under "Tables", many tables are listed, including dbo\_CUSTOMER, dbo\_CUSTOMER1, dbo\_DATI, dbo\_DATI1, dbo\_EMPLOYED, dbo\_EMPLOYED1, dbo\_INVOICE, dbo\_INVOICE1, dbo\_MENU, dbo\_MENU1, dbo\_MOVIE, dbo\_MOVIE1, dbo\_ORDERS, dbo\_ORDERS1, dbo\_SCREEN, dbo\_SCREEN1, dbo\_THEATRE, dbo\_THEATRE1, dbo\_TICKET, and dbo\_TICKET1. Under "Forms", the frmCustomers form is listed. At the bottom, there's a record navigation bar with buttons for first, previous, next, last, and search.

Figure 9.24

The screenshot shows a form titled "Ticket". It has five text boxes: TICKET\_ID (100047), CUSTOMER\_ID (000012), MOVIE\_ID (A10018), DATE\_TIME\_ID (20120404113223), and SEAT (900028). The top of the window has tabs for "frmTicket", "frmMovie\_Sales", "frmMovie1", "frmMovie\_Sales1", "Report1", and "PrintPreview". At the bottom, there's a record navigation bar with buttons for first, previous, next, last, and search.

Form for new ticket sales

Figure 9.25

The Avengers Ticket Sales

Movie Title	Number of Tickets
THE AVENGERS	40
THE AVENGERS: AGE OF ULTRON	40
THE AVENGERS: INFINITY WAR	40
	120

Sample Report for individual movies or movies series sales summaries

Create a report for one of your tables or for a view in Access. Use the Building Access Reports tutorial located on Canvas. Your report must have either group totals or report totals or both.

Figure 9.27

Take a snapshot of your report (with data). (CHRISTINE BECK - Sales by Department)

Sales by Department				
CHECK_TYPE	EMPLOYEE_ID	TOTAL	DISCOUNT	TIP
BAR	000040	2	0	0
		<b>Sum</b>	<b>2</b>	<b>0</b>
<i>Summary for [CHECK_TYPE = BAR] (4 detail records)</i>				
SNACK STAND	000001	12	0	2.4
	000002	7	0	1.4
	000003	8	0	1.6
	000004	27	2.7	5.4
	000005	5.5	0	1.1
	000006	6	0	1.2
	000007	13	0	2.6
	000008	7	0	1.4
	000009	15	1.5	3
	000014	7	0	1.4
	000015	7.5	0	1.2
	000016	16	1.6	3.2
	000017	7	0	1.4
	000018	7.5	0	1.2
	000019	8	0	1.6
	000020	21	2.1	4.2
	000021	30	3	6
	000022	8	0	1.6
	000023	7	0	1.4
	000024	30	3	6
	000025	8	0	1.6
	000026	12	0	2.4
	000027	4	0	0.8
	000028	8	0	1.6
	000029	4	0	0.8
	000030	4	0	0.8
	000031	20	2	4
	000032	5	0	1
	000033	15	1.5	3
	000034	0.5	0	0
000035	1	0	0	
000036	1.5	0	0	
000037	6	0	1.2	
000038	0.5	0	0	
000039	1	0	0.2	
<b>Sum</b>				
<i>Summary for [CHECK_TYPE = SNACK STAND] (35 detail records)</i>				
TICKET COUNTER	000010	11	0	2.2
	000011	28	2.8	5.6
	000012	12	0	2.4
	000013	39	3.9	7.8
<b>Sum</b>				
<i>Summary for [CHECK_TYPE = TICKET COUNTER] (4 detail records)</i>				
<b>Grand Total</b>				
		<b>432</b>	<b>24.1</b>	<b>84.7</b>

Saturday, April 16, 2022
Page 1 of 2
Saturday, April 16, 2022
Page 2 of 2

Figure 9.28

**PALERMO THEATRE INVOICE**

Sunday, April 24, 2022  
5:31:51 PM

Invoice Number	Server
<b>New Invoice</b>	
Department	Ticket Purchase
Menu ID	Inventory ID
<b>TOTAL</b>	
DISCOUNT	
TAX	TIP

---

Invoice Num	TIP	Department	TICKET_ID	MENU_ID	INVENTORY	TOTAL	DISCOUNT	TAX	EMPLOYED_
200035	000039	SNACK STAND	600035	533	1	0	0.06	0.2	
200036	000040	BAR	600036	400	2	0	0.12	0	
200037	000031	BAR	600037	410	5	0	0.3	1	
200038	000032	BAR	600038	420	10	0	0.6	2	
200039	000041	SNACK STAND	600039	580	7	0	0.42	1.4	
200040	000041	SNACK STAND	600040	581	7	0	0.42	1.4	
200041	000041	SNACK STAND	600041	582	2	0	0.12	0	
200042	000041	SNACK STAND	600042	590	2	0	0.12	0	
200043	000041	SNACK STAND	600043	595	2	0	0.12	0	
200044	000041	SNACK STAND	600028	550	5	0	0.3	1	
200045	000041	SNACK STAND	600019	730	7	0	0.4	2	
200047	000001	SNACK STAND	600014	711	13	0	0.73	3	
*									

Form for creating new invoices for all future sales

Figure 9.29

<b>Palermo's Top-Selling Movies</b>	
Movie Title	Number of Tickets
STAR WARS EPISODE VIII	4
HARRY POTTER & DEATHLY HALLOWS P1	4
TITANIC	3
HARRY POTTER & THE SORCERERS STONE	2
KILL BILL: VOL 1	1
JURASSIC PARK	1
JAWS	1
INGLORIOUS BASTERDS	1
HARRY POTTER & PRISONER OF AZKABAN	1
HARRY POTTER & ORDER OF THE PHOENIX	1
LION KING	1
HARRY POTTER & GOBLET OF FIRE	1
SNOW WHITE & THE SEVEN DWARVES	1
HARRY POTTER & DEATHLY HALLOWS P2	1
HARRY POTTER & CHAMBER OF SECRETS	1
GHOST	1
BACK TO THE FUTURE P3	1
BACK TO THE FUTURE P2	1
HARRY POTTER & HALF BLOOD PRINCE	1
STAR WARS EPISODE VI	1
TOY STORY 3	1

Movie Title	Number of Tickets
TOY STORY 2	1
TOY STORY	1
THE AVENGERS: INFINITY WAR	1
THE AVENGERS: END GAME	1
THE AVENGERS: AGE OF ULTRON	1
ROCKY	1
STAR WARS EPISODE VII	1
BACK TO THE FUTURE	1
STAR WARS EPISODE IX	1
STAR WARS EPISODE III	1
STAR WARS EPISODE II	1
STAR WARS EPISODE I	1
STAR WARS	1
TOY STORY 4	1
SLEEPING BEAUTY	1
THE AVENGERS	1
<b>Total Number of Tickets Sold</b>	<b>46</b>

Report showing Palermo's Top-Selling Movies

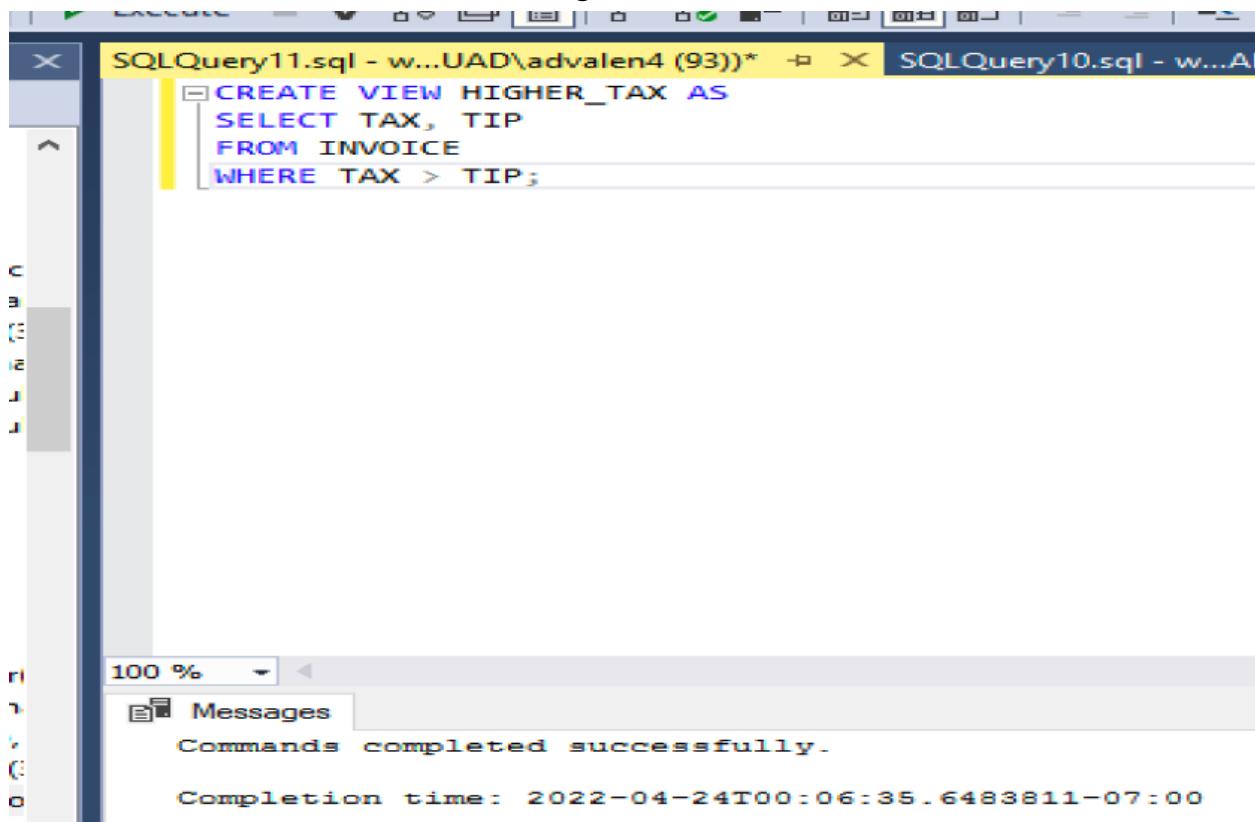
# TABLEAU VISUALIZATION

Created View Scripts and Tableau Visualizations to match:

View #1 Script and Visualization:

```
create view HIGHER_TAX as  
select TAX, TIP  
from INVOICE  
where TAX > TIP;
```

Figure 10.1



The screenshot shows the SQL Server Management Studio interface with two query windows open. The left window, titled 'SQLQuery11.sql - w...UAD\advalen4 (93)\*', contains the SQL script for creating the view:

```
CREATE VIEW HIGHER_TAX AS  
SELECT TAX, TIP  
FROM INVOICE  
WHERE TAX > TIP;
```

The right window, titled 'SQLQuery10.sql - w...A...', is visible but empty. At the bottom of the screen, the 'Messages' pane displays the output of the command:

```
Commands completed successfully.  
Completion time: 2022-04-24T00:06:35.6483811-07:00
```

Successful Creation of HIGHER\_TAX View

Figure 10.2

The screenshot shows a SQL Server Management Studio (SSMS) interface. The title bar of the query window is labeled "SQLQuery12.sql - w...UAD\advalen4 (63)\*". The query itself is:

```
SELECT *  
FROM HIGHER_TAX;
```

The results pane below shows a table with two columns: TAX and TIP. The data is as follows:

	TAX	TIP
1	0.03	0.00
2	0.06	0.00
3	0.06	0.00
4	0.03	0.00
5	0.12	0.00
6	0.12	0.00
7	0.12	0.00
8	0.12	0.00

Contents of HIGHER\_TAX View

Figure 10.3

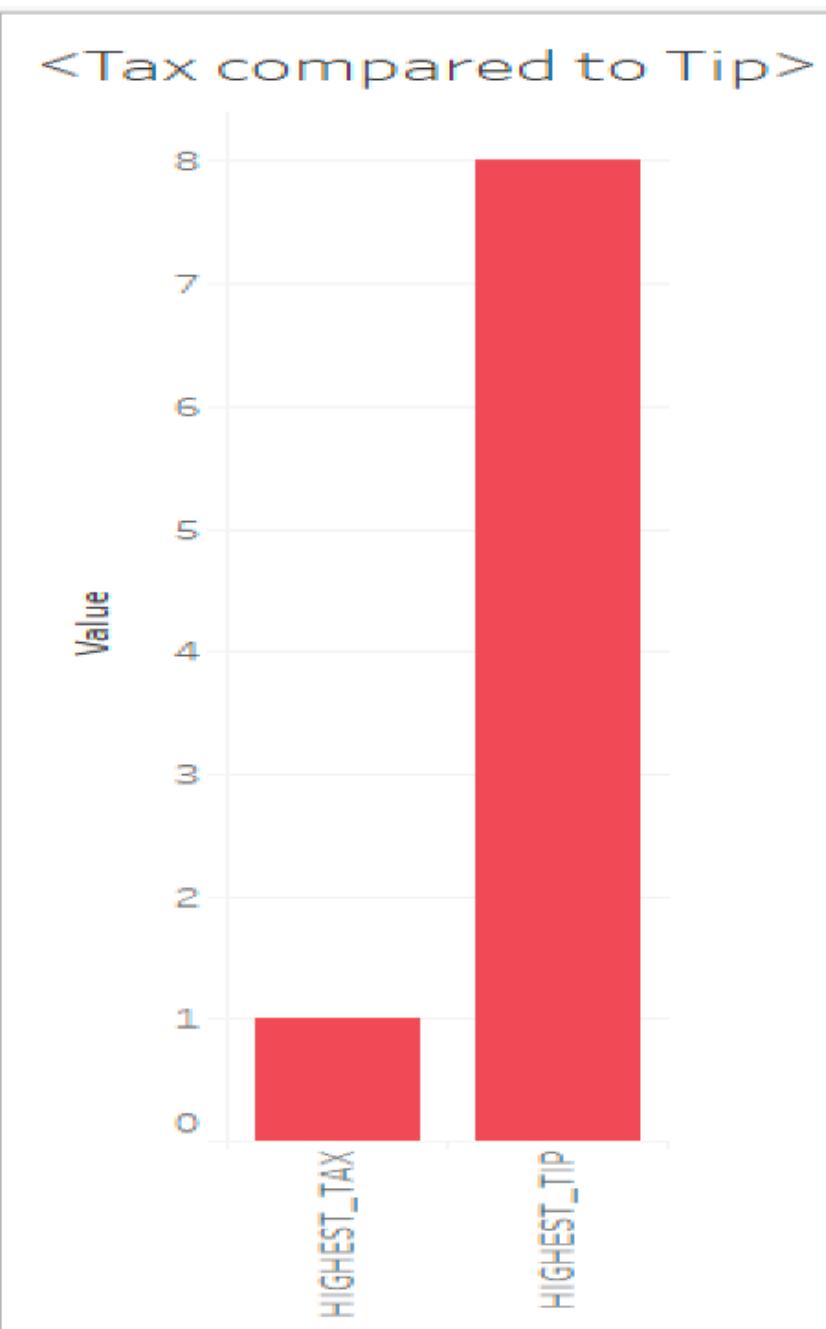


Tableau Visualization of HIGHER\_TAX View

View #2 Script & Visualization:

```
create view RewardLevel1 as  
select *  
from CUSTOMER
```

Figure 10.4

The screenshot shows a SQL query window titled "SQLQuery1.sql - wp...UAD\cigarci4 (59)\*". The query is:`CREATE VIEW RewardLevel1 as  
select *  
from CUSTOMER ;`

Below the query, the status bar shows "80 %". In the "Messages" pane, it says "Commands completed successfully." and "Completion time: 2022-04-22T21:09:43.6116901+09:00".

Successful Creation of RewardLevel1 View

Figure 10.5

Reward Level Data

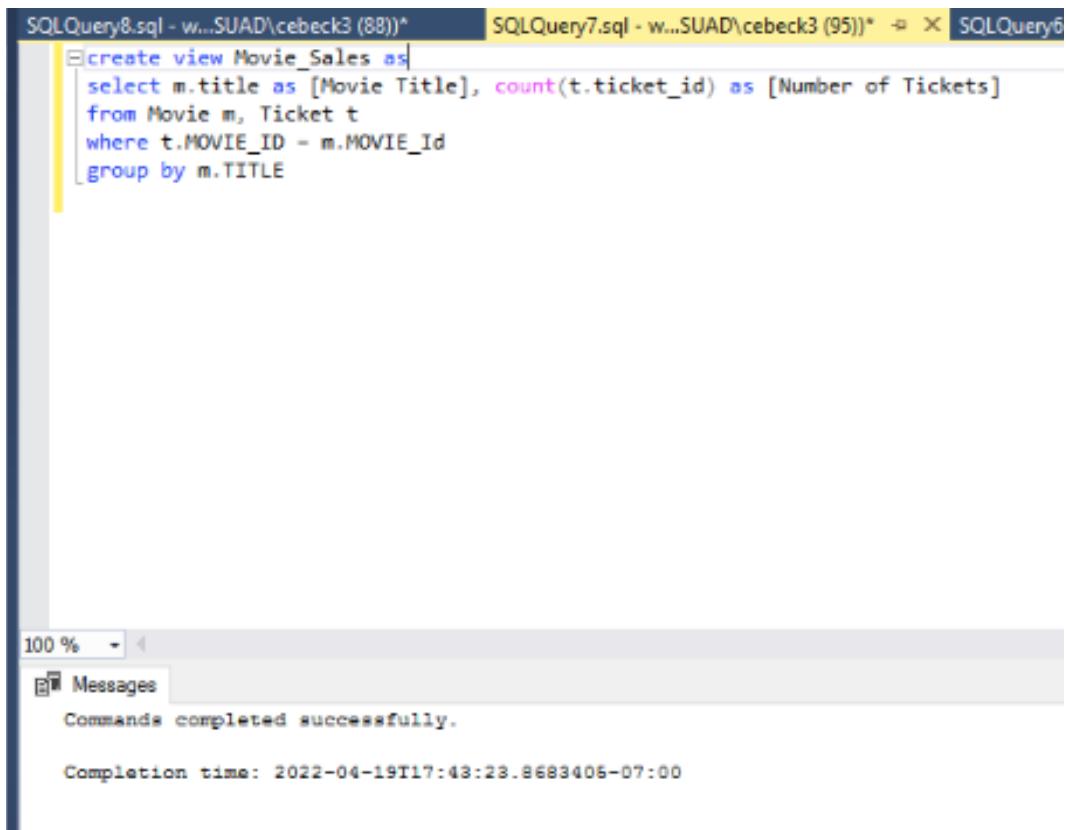
Rwd Lvl Id	
EMMY	21
GRAMMY	24
OSCAR	10
TONY	1

Tableau Visualization of RewardLevel1 View before Triggers & Procedures  
from Appendix A added to SQL

View #3 Script & Visualization:

```
create view Movie_Sales as
select M.TITLE as [Movie Title], count(T.TICKET_ID) as [Number of Tickets]
from MOVIE M, TICKET T
where T.MOVIE_ID = M.MOVIE_ID
group by M.TITLE
```

Figure 10.6



The screenshot shows a SQL Server Management Studio (SSMS) interface with three tabs at the top: 'SQLQuery8.sql - w...SUAD\cebeck3 (88)\*' (highlighted in yellow), 'SQLQuery7.sql - w...SUAD\cebeck3 (95)\*', and 'SQLQuery6'. The main window displays the SQL script for creating the 'Movie\_Sales' view. The script is as follows:

```
create view Movie_Sales as
select m.title as [Movie Title], count(t.ticket_id) as [Number of Tickets]
from Movie m, Ticket t
where t.MOVIE_ID = m.MOVIE_ID
group by m.TITLE
```

Below the script, the 'Messages' pane shows the execution results:

- Commands completed successfully.
- Completion time: 2022-04-19T17:43:23.8683405-07:00

Successful Creation of Movie\_Sales View

Figure 10.7

SQLQuery1.sql - w...SUAD\cebeck3 (56) \* → X

select \* from Movie\_Sales order by 2 desc

100 % ←

Results Messages

	Movie Title	Number of Tickets
1	HARRY POTTER & DEATHLY HALLOWS P1	4
2	STAR WARS EPISODE VIII	4
3	TITANIC	3
4	HARRY POTTER & THE SORCERERS STONE	2
5	INGLORIOUS BASTERDS	1
6	JAWS	1
7	JURASSIC PARK	1
8	KILL BILL VOL 1	1
9	LION KING	1
10	ROCKY	1
11	SLEEPING BEAUTY	1
12	SNOW WHITE & THE SEVEN DWARVES	1
13	STAR WARS	1
14	STAR WARS EPISODE I	1
15	STAR WARS EPISODE II	1
16	STAR WARS EPISODE III	1
17	STAR WARS EPISODE IX	1
18	STAR WARS EPISODE VI	1
19	STAR WARS EPISODE VII	1
20	TOY STORY	1
21	TOY STORY 2	1
22	TOY STORY 3	1
23	TOY STORY 4	1
24	THE AVENGERS	1
25	THE AVENGERS AGE OF ULTRON	1
26	THE AVENGERS END GAME	1
27	THE AVENGERS INFINITY WAR	1
28	HARRY POTTER & DEATHLY HALLOWS P2	1
29	HARRY POTTER & GOBLET OF FIRE	1
30	HARRY POTTER & HALF BLOOD PRINCE	1
31	HARRY POTTER & ORDER OF THE PHOENIX	1

Query executed successfully.

Contents of Movie\_Sales View

Figure 10.8

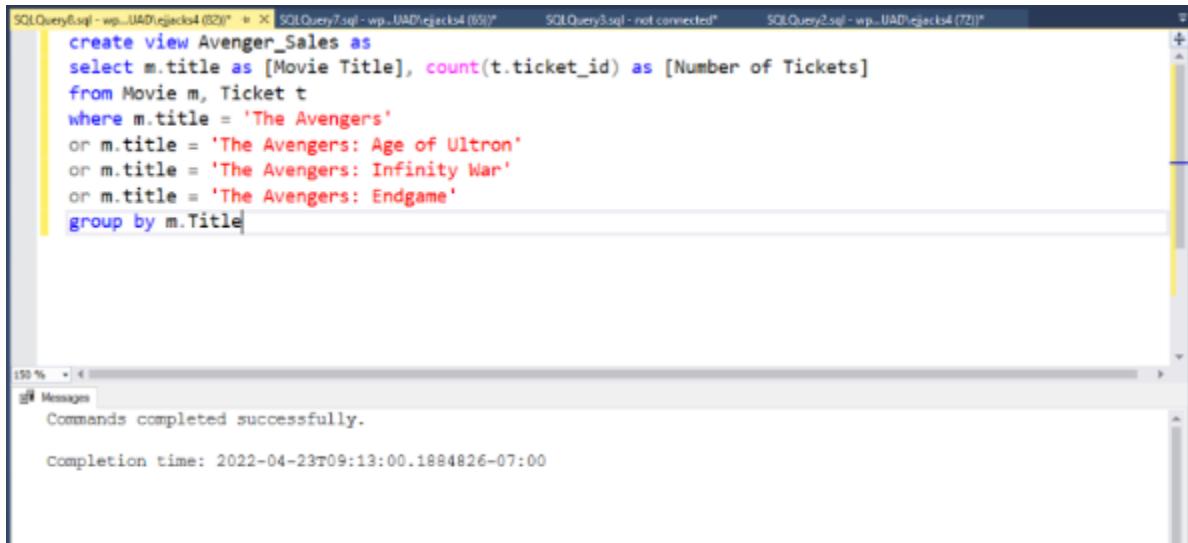


Tableau Visualization of Movie\_Sales View

View #4 Script & Visualization:

```
create view Avenger_Sales as
select M.TITLE as [Movie Title], count(T.TICKET_ID) as [Number of Tickets]
from MOVIE M, TICKET T
where M.TITLE = 'The Avengers'
or M.TITLE = 'The Avengers: Age of Ultron'
or M.TITLE = 'The Avengers: Infinity War'
or M.TITLE = 'The Avengers: Endgame'
group by M.TITLE
```

Figure 10.9

A screenshot of the SQL Server Management Studio (SSMS) interface. The central pane shows a T-SQL script for creating a view named 'Avenger\_Sales'. The script includes a SELECT statement with COUNT(t.ticket\_id) as 'Number of Tickets' from 'MOVIE M' and 'TICKET T', filtered by movie titles ('The Avengers', 'The Avengers: Age of Ultron', 'The Avengers: Infinity War', or 'The Avengers: Endgame') and grouped by movie title. The status bar at the bottom indicates the command was completed successfully at 2022-04-23T09:13:00.1884826-07:00.

```
create view Avenger_Sales as
select m.title as [Movie Title], count(t.ticket_id) as [Number of Tickets]
from Movie m, Ticket t
where m.title = 'The Avengers'
or m.title = 'The Avengers: Age of Ultron'
or m.title = 'The Avengers: Infinity War'
or m.title = 'The Avengers: Endgame'
group by m.Title

# Messages
Commands completed successfully.

Completion time: 2022-04-23T09:13:00.1884826-07:00
```

Successful Creation of Avenger\_Sales View

Figure 10.10

The screenshot shows a SQL Server Management Studio window with two panes. The left pane contains a query editor with the following code:

```
SQLQueryLog - wp...UAD\jacks4 (82) | SQLQuery7.sql - wp...UAD\jacks4
select *
from Avenger_Sales
```

The right pane is titled "Results" and displays the output of the query as a table:

	Movie Title	Number of Tickets
1	THE AVENGERS	40
2	THE AVENGERS: AGE OF ULTRON	40
3	THE AVENGERS: INFINITY WAR	40

Contents of Avenger\_Sales View

Figure 10.11

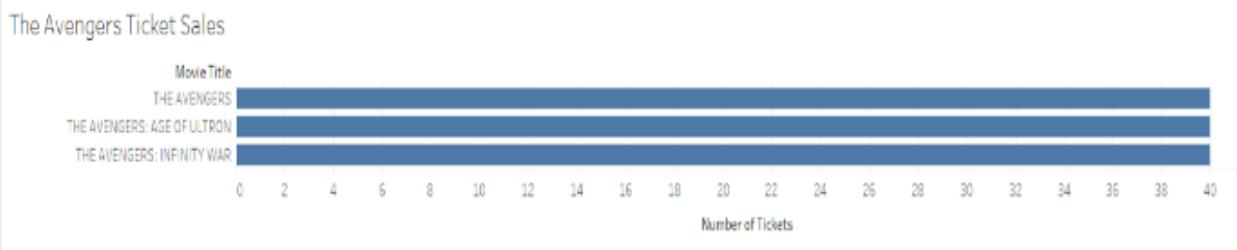


Tableau Visualization of Avenger\_Sales View

View #5 Script & Visualization:

```
create view [Jacqueline Qwerty] as  
select E.F_NAME, E.L_NAME, E.EMPLOYED_ID, I.INVOICE_ID  
from EMPLOYED as E, INVOICE as I  
where E.EMPLOYED_ID = I.EMPLOYED_ID and E.EMPLOYED_ID = 000041
```

Figure 10.12

The screenshot shows the SQL Server Management Studio interface. In the top tab bar, there are two tabs: "SQLQuery137.sql - not connected\*" and "SQLQuery148.sql - ...SUAD\ejmoss1 (90)\*". The main window displays the T-SQL script for creating a view:

```
CREATE VIEW [Jacqueline Qwerty] AS  
SELECT E.F_NAME, E.L_NAME, E.EMPLOYED_ID, I.INVOICE_ID  
FROM EMPLOYED AS E, INVOICE AS I  
WHERE E.EMPLOYED_ID = I.EMPLOYED_ID AND E.EMPLOYED_ID = 000041
```

Below the script, the "Messages" pane shows the execution results:

```
Commands completed successfully.  
Completion time: 2022-04-23T09:28:57.3108065-07:00
```

At the bottom of the window, the status bar indicates:

```
150 % | wpcmssql.cloud.wpcarey.asu.... | SUAD\ejmoss1 (90) | BonlineB360 | 00:00:00 | 0 rows
```

Successful Creation of Jacqueline Qwerty View

Figure 10.13

select * from [Jacqueline Qwerty]				
100 %				
	Results	Messages		
1	JACQUELINE	QWERTY	000041	200039
2	JACQUELINE	QWERTY	000041	200040
3	JACQUELINE	QWERTY	000041	200041
4	JACQUELINE	QWERTY	000041	200042
5	JACQUELINE	QWERTY	000041	200043
6	JACQUELINE	QWERTY	000041	200044
7	JACQUELINE	QWERTY	000041	200045

Contents of Jacqueline Qwerty View

Figure 10.14

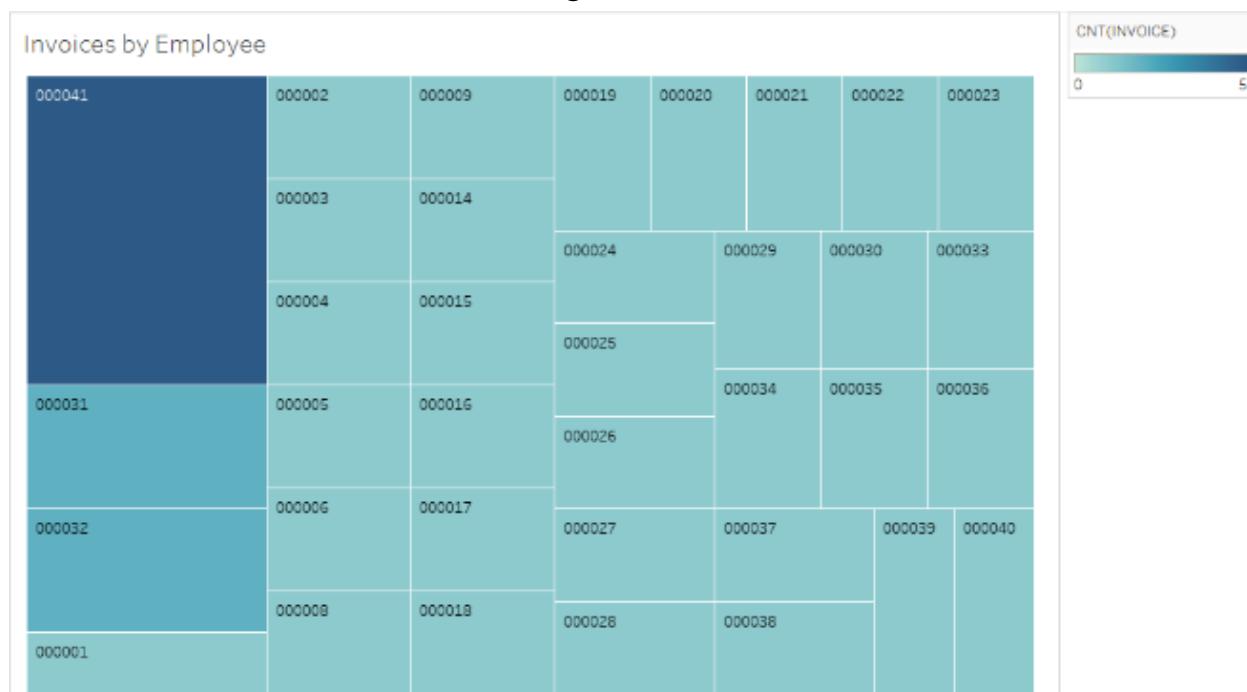


Tableau Visualization of Jacqueline Qwerty View

View #6 Script & Visualization:

```
create view MovieRatings as
select M.RATING as 'Movie Rating', count(M.RATING) as 'Number of Movies With Rating'
from MOVIE M
group by M.RATING
```

Figure 10.15

The screenshot shows a database interface with a SQL editor window and a messages window.

In the SQL editor window, the following SQL script is displayed:

```
create view MovieRatings as
select m.rating as 'Movie Rating', count(m.rating) as 'Number of Movies With Rating'
From movie m
group by m.RATING
```

In the messages window, the output is:

Commands completed successfully.

Completion time : 2022-04-24T14:11:01.2883200-07:00

Successful Creation of MovieRatings View

Figure 10.16

The screenshot shows a SQL Server Management Studio (SSMS) interface with two tabs open. The left tab is titled 'SOLIDquery.sql - wp...SUMD(yesnoall) (76)' and contains the following SQL code:

```
select *
from MovieRatings
```

The right tab is titled 'SOLIDquery.sql - wp...SUMD(yesnoall) (91)' and displays the results of the executed query. The results are presented in a table with two columns: 'Movie Rating' and 'Number of Rows With Rating'. The data is as follows:

Movie Rating	Number of Rows With Rating
1	7
2	17
3	19
4	3

Contents of MovieRatings View

Figure 10.17



Tableau Visualization of MovieRatings View

View #6 Script & Visualization:

```
create view [RESTOCKS] as  
select * from INVENTORY  
where QOH < RESTOCK_LEVEL
```

Figure 10.18

The screenshot shows a SQL Server Management Studio (SSMS) interface. On the left, there's a tree view of database objects. In the center, two query panes are open: 'SQLQuery7' and 'SQLQuery14'. The 'SQLQuery7' pane contains the T-SQL script for creating the 'RESTOCKS' view. The 'Messages' tab at the bottom shows a successful execution message.

```
CREATE VIEW [RESTOCKS] AS  
SELECT * FROM INVENTORY  
WHERE QOH < RESTOCK_LEVEL
```

Messages

Commands completed successfully.

Completion time: 2023-04-24T11:04:23.8944331+08:00

Successful Creation of RESTOCKS View

Figure 10.19

The screenshot shows a SQL query results window. The query is:

```
SELECT * FROM [restocks]
```

The results are displayed in a table with the following columns: INVENTORY\_ID, RESTOCK\_LEVEL, ITEM\_DESC, SERVING\_SIZE, QOH, and QOO. The data consists of 15 rows, each representing a different item with its details.

	INVENTORY_ID	RESTOCK_LEVEL	ITEM_DESC	SERVING_SIZE	QOH	QOO
1	410	000	WINE	5 OZ	705	0
2	420	000	COCKTAIL	1.5 OZ	723	0
3	531	300	VEGGIE STICKS W/ HUMMUS	NULL	250	0
4	532	50	SIDE OF DRESSING	NULL	42	0
5	533	50	SIDE OF HUMMUS	NULL	31	100
6	552	50	SIDE OF CHEESE	NULL	22	0
7	553	50	SIDE OF MUSTARD	NULL	11	100
8	595	250	VEGGIE BURGER W/ CHEESE	NULL	247	0
9	600	500	M&Ms	NULL	41	0
10	601	500	PEANUT M&Ms	NULL	232	100
11	701	50	10 INCH CHEESE PIZZA	10 IN	45	0
12	800	700	SMALL POPCORN	SMALL	687	0
13	801	700	MEDIUM POPCORN	MED	688	0
14	802	700	LARGE POPCORN	LARG	688	0
15	813	700	EXTREME LARGE POPCORN	XL	688	0

Query executed successfully.

wpesmq.cloud.wpcarey.edu... ASUAD\zdliewis6 (103) | Business360 | 00:00:00 | 16 rows

Ln 1 Col 25 Ch 25 INS

Contents of RESTOCKS View

Figure 10.20

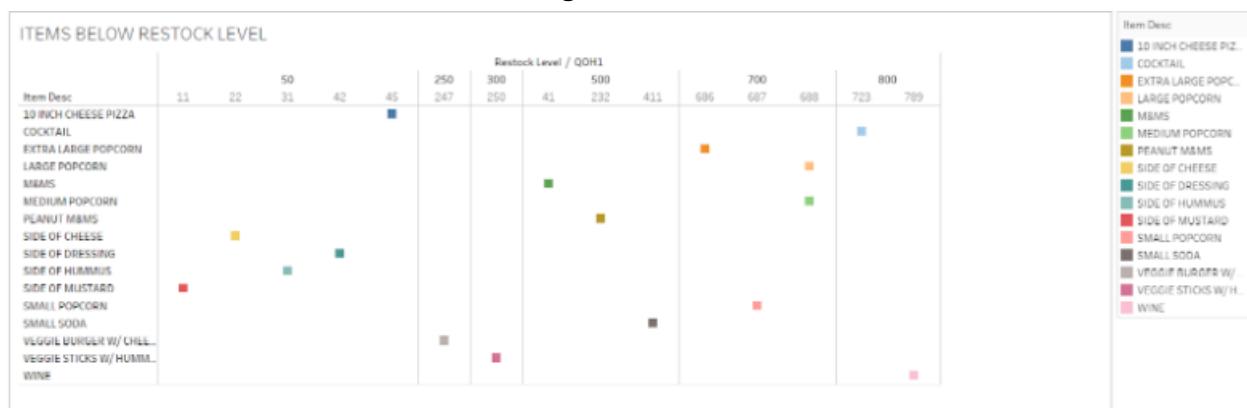


Tableau Visualization of RESTOCKS View

# APPENDICES B: TEAM MEMBER RESUMES AND TIME SHEETS

---

## CHRISTINE E. BECK

---

Flagstaff | Phoenix, Arizona

P: 928.864.8123

E: cebeck816@gmail.com

### EXPERIENCE

ACCOUNTING ASSISTANT, Compass Point Accounting, PLLC — February 2013 to Present

Provide bookkeeping and payroll services for Arizona businesses. Collaborate directly with firm CPA and clients on financial issues. Assist with the preparation of business and personal tax returns. Experience in this role includes QuickBooks accountant software, QuickBooks Online bookkeeping, Lacerte tax preparation software, W2 Mate tax form software, and Wave online bookkeeping.

CO-OWNER | CONTROLLER, Flagstaff Hobbies, Inc. — October 2008 to August 2021

Successfully operated and sold a hobby and game retailer. Managed supply chain, supplier relations, inventory purchasing/receiving and product placement. Oversaw online and event marketing efforts. Prepared and maintained business accounting records including budgets, payables and reconciliations. Experience in this role included QuickBooks Online and QuickBooks point of sale.

ACCOUNTING ASSISTANT, Robert Russel, CPA — May 2011 to January 2013

Provided bookkeeping and payroll services for Arizona businesses. Experience in this role included Peachtree bookkeeping software.

### PROFICIENCIES STRENGTHS

I am a fast learner. Under a skilled manager, I effectively oversee my workload and complete tasks by their given deadlines. I am a team member who requires minimal supervision once my responsibilities are outlined.

- Microsoft Office
- Google Suite
- Quickbooks
- Zoom
- Leadership
- Organization
- Marketing/Event Planning
- Time Management
- Communications

### EDUCATION

Arizona State University, Bachelors in Accountancy Program (GPA 4.0) — 2021 to Present

Beta Gamma Sigma International Business Honor Society — 2022

Coconino Community College, Associates in Business (GPA 3.87, Magna Cum Laude)  
— Class 2011

Phi Theta Kappa International Honor Society, Beta Gamma Chi Chapter — 2008

VP of Scholarship — 2009 to 2011

## **Jorge Castro**

(480) 544-3241 - JECASILLASCASTRO@GMAIL..COM

### **Skills**

- Bilingual ( English, Spanish)
- Willing to seek out information to problem solve
- Great teamwork skills
- Ability to speak in front of large groups
- Able to work independently

### **Experience**

May 2021 - March 2022

#### **Twisted Grove, Scottsdale - Bartender/Server**

- Opened and closed bar in a timely manner
- Served alcoholic and nonalcoholic beverages to guests in a timely professional manner
- Utilized knowledge of mixed drinks and cocktails to create beverages for featuring
- Solely made mixed drinks for all of restaurant on weekends
- Stocked bar and garnishes when running low
- Recommended beverages based on knowledge of bar menu
- Ensured to identify patrons were of legal age to consume alcoholic beverages

JULY 2018 - May 2021

#### **Starbucks, Tempe - Shift Supervisor**

- In depth handling of customer relations
- Interviewed and hired applicants
- Assessed baristas abilities in order for them to promote
- Ensured daily and weekly orders were placed in a timely manner
- Managed supplies to ensure proper inventory at all times
- Lead training for new shift supervisors
- Evaluated peers on performance
- Led Shift Supervisor meetings to discuss potential points of improvement

JUNE 2013 - JUNE 2018

#### **United States Marine Corps, Hawaii - Infantrymen/ Rifleman**

- Assisted in making decisions for daily tasks
- Ability to work well and communicate efficiently with foreign nationals
- Coordinated people for effective performance
- Coordinated daily operations
- Constructively criticized junior Marines
- Evaluated performance of peers
- Performed roles as Fire Team Leader, Squad Leader and Platoon Sergeant.

Education

JANUARY 2019 - Present

**Arizona State University, Tempe, Arizona - Accountancy**

JANUARY 2019 - Present

**Chandler-Gilbert Community College, Chandler, Arizona - General Education**

AUGUST 2009 - JUNE 2013

**Lakota East High School, Liberty Township, Ohio - High School Diploma**

**Celeste Garcia**

cigarci4@asu.edu

---

2332 West Corral Road, Phoenix, AZ 85041

(520) 668-6299

## **EDUCATION**

W. P. Carey School of Business, Arizona State University, Tempe, AZ May 2023

Bachelor of Science, Accountancy

Korean Studies Certificate, International Business Certificate.

- Awards: Obama Scholar, New American University Scholarship
- GPA: 3.95

## **PROFESSIONAL EXPERIENCE**

Discover card, Education at Work, Account Manager.

December 2019 – March 2020

Phoenix, AZ

- Launched outbound calls to customers to gain cell consent. Transferring calls to a collection specialist for further assistance.
- Maintained productivity standards and performed quality collection work with timely follow up.
- Engages in team building activities among colleagues.

Wildflower Bread Company, Cashier/Busser.

August 2019 – December 2019

Chandler, AZ

- Collaborated with colleagues in bussing tables.
- Handled cash, credit, or check transactions with customers. Issue refunds to necessary transactions.

Service Systems Associates, Reid Park Zoo, Cashier/Trainer/Catering Team.

August 2016 – July 2019

Tucson, AZ

- Trained and mentored new employees by facilitating various multiple day trainings.
- Administered cash, credit, or check transactions with customers. Issue refunds to necessary transactions.
- Event set up, shut down and event schedule management.
- Application of customer service skills during every customer transaction.

ASU Financial Aid and Scholarship Services, Student aid representative

June

2021-Present

Phoenix, AZ

- Assisted and addressed students and customers via phone, email or in person with questions pertaining to financial aid.
- Utilizing PeopleSoft, Phone program and Arizona State University Website.

ASU SEMTE Business Operations Office Aid

June

2021-Present

Phoenix, Az

- Clerical Tasks; Front Desk Rep, Data entry, Phone service, electronic filing, etc.

## **SKILLS & ACTIVITIES**

Technical/Soft Skills: Beginning Excel, interpersonal skills, organizational skills, adaptability, etc.

Languages: English and Korean (Intermediate)

Volunteer: ITMHCA

Leadership: GAP Inc. Learning Experience.

# Elijah J. Jackson

ejjacks4@asu.edu | (209) 684-5149 | www.linkedin.com/in/elijah-jackson4

## OBJECTIVE

To seek out internships or opportunities to start my career and begin practicing accounting principles in a dynamic and growing business environment.

## EDUCATION

W. P. Carey School of Business, Arizona State University, Tempe, AZ May 2023

Bachelor of Science, Accountancy

- Awards: Dean's List, New American University Scholarship
- Cumulative GPA: 3.25

## PROFESSIONAL EXPERIENCE

**Line/Prep Cook/Keyholder**, Elbows Mac 'N Cheese, Chandler, AZ

July 2021 – Present

- Train and mentor new employees by demonstrating recipes and preparation techniques
- Operate kitchen as lead cook and delegate tasks to other cooks in order to run kitchen efficiently

**Team Leader**, Harkins Theaters, Gilbert, AZ May

2019 – July 2021

- Delegated tasks to a team to keep area running smoothly
- Operated concession stand, box office, and usher areas of the movie theater
- Trained and mentored new employees by demonstrating tasks and procedures

**Dishwasher**, Otaku Sushi, Chandler, AZ August

2018 – March 2019

- Washed dishes and kept kitchen clean throughout the night
- Helped line cooks prepare and cook food when help was needed
- Ensured kitchen was clean and ready for close nightly

## SKILLS & ACTIVITIES

- Microsoft Office
- Preparing financial statements
- Organization skills
- Strong critical thinking skills
- Strong collaboration skills

## RYU LEWIS

---

Email: [delraylewis@gmail.com](mailto:delraylewis@gmail.com) | LinkedIn: <https://www.linkedin.com/in/ryulewis/>

### EDUCATION

---

#### Arizona State University

*Bachelor of Science, Accountancy*

**Tempe, Arizona**

*Expected: December 2023*

- Cumulative GPA: 3.59
- Dean's List: Fall 2020, Spring 2021, and Fall 2021
- National Association of Black Accountants, Information Technology Gurus of ASU, and W.P. Carey OBSA

### PROJECT / VOLUNTEER EXPERIENCE

---

#### Entrepreneurship Business Pitch

*WPC 101: Student Success in Business*

**Arizona State University**

*September 2019 - December 2019*

- Collaborated with 3 team members to devise a business plan that offered a food delivery service to students in college dorms
- Presented the business plan with a 10 minute oral pitch and utilized a PowerPoint presentation to implement relevant visuals

#### School Improvement Project

*Volunteer*

**CAJ, Japan and Chiang Mai, Thailand**

*March 2019*

- Collaborated with peers to acquire funds for building projects in Northern Thailand that provided opportunities for 100+ hill-tribe school children
- Accomplished a trip to Thailand and worked closely with local organizations to renovate schools

### WORK EXPERIENCE

---

#### The FamilyMart Company, Ltd.

*Convenience Store Clerk*

**Okinawa, Japan**

*August 2016 - March 2017*

- Demonstrated customer service skills by connecting with 50-100 customers per day and advising on optimal product purchases
- Established ideal accounting by calculating daily sales amounting to \$1500-2000, balancing 4 cash drawers, maintaining records, and submitting receipts to management
- Managed the distribution of 300+ units of inventory per day, maintaining a high degree of accuracy to eliminate sales interruptions and customer inconvenience

### SKILLS AND INTERESTS

---

**Computer Skills:** Microsoft Office (Word, Excel, PowerPoint, and Access), and Google Workspace (Gmail, Drive, and Docs)

**Interests:** Traveling, Skiing, Snowboarding, F1 Racing, Soccer, and Contemporary art

# Erika J. Moss

ejmoss1@asu.edu

[www.linkedin.com/in/erika-moss2020](https://www.linkedin.com/in/erika-moss2020)

## EDUCATION

**W. P. Carey School of Business, Arizona State University, Tempe, AZ** December 2022

Bachelor of Science, Accountancy

## PROFESSIONAL EXPERIENCE

**Intern**, CliftonLarsonAllen, Phoenix, AZ June 2021 - August 2021; January 2022 - Present

- Participated in year-end audits and preliminary audit work for state and local government entities such as cities, towns, and school districts
- Performed audit testwork for state compliance such as USFR/HURF, and federal compliance single audit programs
- Communicated with clients via email and online portals for audit evidence requests including selecting samples from reports the client provided and following up with clients regarding questions and findings
- Worked within audit teams to complete preliminary work including updating workpapers
- Completed detailed audit work involving testing of General Ledger items, cash receipts, disbursements, and attendance for K-12 districts.

**Intern**, Heinfeld, Meech, & Co., Scottsdale, AZ September 2021 – November 2021

- Gained skills to complete year-end audits and audit work for clients in the state and local government sector
- Worked on an audit team to complete workpapers, review documentation, and become familiar with state and federal regulations
- Performed testwork to ensure compliance with federal and state programs

**Head Assistant & Tutor**, Kumon Learning Academy, Chandler, AZ

Tutor: February 2016 – August 2017

Head Assistant: April 2018 – November 2018

- Tutored reading and math to an average of 15 children of all ages each day
- Graded homework and classwork after each class period in order to ensure that students understood the class material and that the learning needs of each individual student had been met
- Prepared classwork, homework, filed folders, and took inventory of learning materials for the next day's class in order to guarantee that class flows better the next day
- Interfaced with parents weekly to inform them of their child's progress and what was worked on or needs to be worked on
- Trained new tutors and oversaw the work of an average of five tutors at a time

# Alec Valencia

Goodyear, AZ | Advalen4@asu.edu | (623)-302-0470 | [Linkedin.com/alec-valencia-5489381a3](https://www.linkedin.com/in/alec-valencia-5489381a3)

## EDUCATION

### Arizona State University

**W.P. Carey School of Business** | *Bachelor of Arts, Information Security*

Tempe, AZ | **08/2018 -**

**05/2022**

### Agua Fria High School

Avondale, AZ |

**08/2013- 05/2017**

## PROGRAMMING SKILLS

- **Programming languages:** Python, SQL, JavaScript, and HTML **Technical skills:** Advanced
- **Software tools:** Visual studio, JMP, Tableau **Technical skills:** Advanced
- **Security tools:** Kali linux, Parrot OS, Linux, Mac Advanced OS, Powershell, and WireShark **Technical skills:**
- **Communication:** Written & Verbal Communication: **Language proficiency:** Fluent English

## EXPERIENCE

### Computer Programmer | Freelance

Goodyear, AZ | **08/2019 -**

**Present**

- Coded a python script to accurately track whether a product was in stock or out of stock
- Organized, encrypted and decrypted information
- Create passwords and or create security measures and protocols

### Computer repair technician | Self-employed

Goodyear, AZ | **07/2018 -**

**Present**

- Build PC tower from scratch with individual parts
- Virus, malware removal and prevention
- Software application errors and computer startup issues support

# TEAM JOURNAL ENTRIES

<b>Team Member Name:</b>	<b>Alec Valencia</b>	
Date of work performed	Description of work performed	Time Spent on Work Performed
3/26/2022	Team Meeting	1.5
3/23/2022	Intro paragraph	1
3/25/2022	Team logo	1
3/28/2022	Team Meeting	1
3/30/2022	ERD	2
3/31/2022	Data Dictionary	2
4/2/2022	Team Meeting	1
4/5/2022	Team Meeting/Scripts	2
4/7/2022	Tables and Queries	5
4/11/2022	Team meeting	4
4/12/2022	Team Meeting & Inserting data	3

4/13/2022	Assigned queries to team	2
<b>Team Member Name:</b>	<b>Alec Valencia</b>	
4/14/2022	Schema	2
4/15/22	Nested queries & Basic queries	5
4/16/22	Team meeting/hw 5 & hw 6	4
4/17/22	Finalizing schema and making sure queries run	2
4/18/22	Team Meeting	3
4/19/22	Milestone 4	2
4/20/22	Milestone 4 transactions	2
4/21/22	Milestone 4 data visualization	2
4/22/22	Reports & Forms for Milestone 4	2
4/23/22	Team meeting/Milestone 4	5
4/26/22	Milestone 5/ Milestone 6	5
4/28/22	Presentation and inserting data into milestone 6	6
4/29/22	Formatting milestone 6	5

<b>Team member name:</b>	Alec Valencia	
	<b>Total Hours:</b>	70.5

<b>Team Member Name:</b>	<b>Celeste Garcia</b>	
<b>Date of work performed (*Dates according to South Korean KST Time. +16 hours from MST*)</b>	<b>Description of Work Performed</b>	<b>Time Spent on Work Performed (Hrs)</b>
3/21	Team meeting for milestone 1	1
3/22	Introductory Paragraph and Resume	1
3/26	Team meeting to finalize submission for M1. +Submission of document	0.4
3/29	Milestone meeting, collaborated in working on creating Data Dictionary and ERD.	2
3/31	Added further to Data Dictionary	0.5
4/3	Finalized Data Dictionary and ERD, ensured they matched each other.	1
4/5	group meeting	0.5
4/6	Working on scripts	1
4/10	Team meeting, adding initial scripts prior to review of professor suggestion	4.5
4/11	Writing scripts	2

<b>Team Member Name: Celeste Garcia</b>		
4/12	Team meeting adding tables and inserts, fixing tables and inserts	3.5
4/13	Meeting to add tables and inserts, add table for missing tables based on professor suggestion	2.5
4/14	Fixing data in Customer table, reviewing tables and inserts, reviewing access	1.5
4/15	Reviewing tables, adding foreign keys, working on HW5Q1	6
4/16	Running Queries for Hw 5 and for M3	1
4/17	Team Meeting, HW 5 and HW 6, Updated ERD	3
4/18	Compare ERD prep for meeting	2.6
4/19	Group meeting	2
4/20	M4 assignment	0.5
4/21	M4 assignment	2.5
4/22	M4 assignment	2.5
4/24	Team Meeting for Milestone 4.	4

<b>Team Member Name:</b> Celeste Garcia		
4/26	Team Meeting for Final project, worked on delegated duties after wards	2.5
4/27	Met with Christine to discuss status on project, worked on SQL view, ACL audits, PPT and Final Paper	3.5
4/28	Team meeting followed by working on final paper duties and time sheets. Worked on PPT	3
4/29	Recorded and added audio to final PPT. Worked on final paper duties. Group Meeting.	2
4/30	Final Group Meeting and wrap up	3.5
TOTAL:		60

<b>Team Member Name:</b> Christine Beck		
<b>Date Work Performed</b>	<b>Description of Work Performed</b>	<b>Time Spent of Work Performed (HRS)</b>
3/16/22	Post initial discussion to coordinate team availability and communication preferences	0.5
3/20/22	Email group about initial discussion (since half did not respond to by 3/20), post discussion about Milestone 1 and possible meeting times to complete assignment, work on personal introductory paragraph for Milestone 1.	1.00
3/21/22	test ASU Zoom meeting tools with friend outside of ASU, create ASU Zoom meeting for group project, invite team, post discussion in Group B for meeting changes., meet with team (erika attended, celeste responded that she couldn't make it, ended meeting at 6:30pm, emailed notes from meeting), arranged another zoom meet for 3/23 & sent invite to team.	2.50
3/22/22	post meeting update in Canvas group discussion, schedule email reminder with same information	0.33

3/23/22	as zoom host prepare for scheduled team meeting, review agenda and expectations with teammates in attendance (Erika, Alec, Celeste, Jorge), delegate duties (celeste will submit, jorge will attend saturday office hours to ask questions of the professor about the project, alec will work on a team logo), schedule another meeting for Saturday 3/26 to finalize milestone 1, email notes (taken by erika) from meeting to group and post in group canvas discussion.	2.00
<b>Team Member Name:</b> Christine Beck		
3/26/22	write up team "sales pitch," scheduled Saturday meeting, schedule Monday zoom meeting, schedule email reminders	1.50
3/28/22	attended scheduled Monday meeting, made good progress on Milestone 2 (all in attendance except Elijah), worked on google sheet for data dictionary, email & post meeting notes	3.00
4/2/22	review Milestone 2 ERD & data dictionary, add comments for changes, email Saturday meeting reminder to group, scheduled Saturday Meeting (all in attendance), complete Milestone 2 discussions, Elijah to submit Milestone 2, schedule Monday meeting reminder email	2.50
4/4/22	scheduled Monday meeting to discuss Milestone 3, delegate table and insert script creation duties, schedule Saturday meeting reminders, start work on my tables, test in my SQL database - not in group	2.50
4/7/22	review changes and make notes on other changes for ERD, email/text/post discussion to group about changes	0.75

4/8/22	scheduled Saturday meeting (all in attendance except Erika & Alec, but they gave notice), review both erd comment emails and all changes to erd, found a few items to address, reassign table scripts for Monday, email meeting notes, schedule Monday reminder email	1.50
<b>Team Member Name:</b> Christine Beck		
4/9/22	review data dictionary, write assigned metadata scripts for theatre and menu item tables	2.50
4/11/22	add more scripts (found where the tables need 40 items) to menu table, run scheduled Monday meeting, work on entering table scripts into sql, have lots of problems, make corrections, finally get all tables entered, insert data for theatre and menu, email for unscheduled meeting for tuesday since we had so many errors tonight. Hopefully can get the rest of the inserts loaded and assign tasks for ALL of the work do Sunday	5.00
4/12/22	another group meeting (all in attendance except for Erika), Alec worked on writing & assigning queries for the group to use to test the database, I helped create inserts for teammates who still had not completed their tasks, fixed errors in movie inserts, discovered an error in DATI table, had to delete the last four tables and tried to recreate, ran into coding issues again - data dictionary, erd, and data scripts are a mess, we've lost all coherency	3.00
4/15/22	schedule Saturday group meeting reminder that outlines what needs to be done before the meeting, send email highlighting issues with database inserts (orders, ticket, invoice). Email professor for guidance about broken data tables. Email team	5.25

	about professor response, complete my M3 Query & View assignment. Complete HW5 Access Report ("Sales by Type").	
<b>Team Member Name:</b> Christine Beck		
4/16/22	meet with professor and erika about diagram, email group about changes and ideas for how we want to continue with group assignments due 4/17. Regularly scheduled Saturday meeting . Update group on findings from morning Professor input. Decide to save the changes for M4 and continue with the group assignments with broken data - it's a lot to recreate before tomorrow's deadlines. Create a second view to use for HW6. Complete HW6 during meeting. Email meeting review and plans for Monday meeting.	4.75
4/17/22	update my assigned metadata scripts to match updated ERD in preparation for database recreation during Monday's group meeting for M4	0.25
4/18/22	delete and recreate database to match weekend input from Professor. Assign tasks for M4 to be completed by Saturday. All in attendance.	2.00
4/19/22	triggers & procedures research, email professor about procedure code, create "movie sales" view, construct "Movie Sales" report	6.00
4/21/22	work with Erika on Trigger inputs for group database	0.50

4/22/22	test access/sql communication updates - there was an easier way to add data to our tables through access but getting an error on fk information, recreate report, create tableau visualizations from movie sales report, attempt invoice form creation - not working to add multiple items to an invoice	2.25
<b>Team Member Name:</b> Christine Beck		
4/23/22	attempt procedure & transactions - getting error about "visit' trigger so it won't let me run new ticket transactions, group meeting (all in attendance), resolve trigger and transaction errors, build dashboard	5.25
4/24/22	complete m4 form (invoice), format & submit M4	5.5
4/25/22	email professor about acl linking, review voicethread requirements before meeting, regularly scheduled group meeting, delegate duties for final paper, presentation, acl audit, email meeting wrapup to group, complete & submit my peer evaluations	4.25
4/26/22	start introduction of presentation slides and voiceover, complete system audit & screenshots, review all project requirements for anything that we may be missing (found some checks from M2 that need to be addressed), email group	5.75
4/27/22	complete introduction slides to presentation, upload to voicethread, record intro, add column to invoice table, add data, complete some of the required queries, start adding m4 screenshots and scripts to final paper, edit some of the paper formatting.	4.75

4/28/22	complete my portion of the final paper - insert all information from M4 (triggers, procedures, views, tableau visualizations - Ryu is in charge of the Access portion), add my ACL test to final paper, work on public link for voicethread, verify with a friend that the public link works, email group that I have completed my steps and will not be able to attend final meetings due to family memorial, but will submit M5 link when everyone has completed their slides and recordings, update timesheet in final paper	4.75
4/29/22	rearrange presentation into proper order, add a couple of the required queries to the final paper, add missing reports and forms that were supposed to be added by a different team member,submit M5	2.25
	<b>Total Hours for Christine Beck:</b>	82.08

Group Member:	<b>Elijah Jackson</b>	
Date of Work Performed	Description of Work Performed	Time Spent on Work Performed (HRS)
3/23	Introductory Paragraph	1
3/23	Team Charter/Code of Ethics	0.25
3/23	Resume	0.5
3/26	Team Meeting	0.33
4/1	Look over and provide suggestions for EERD and schema	0.5
4/2	Team Meeting	1
4/3	Export all files into .pdf and finalized/format final M2 submission	3
4/4	Team Meeting	1
4/6	Wrote table script for tickets	1

4/9	Team Meeting	1.5
4/11	Team Meeting, Write INSERT scripts for ticket	3
Member Name:	<b>Elijah Jackson</b>	2
4/12	Work on SQL tables and inserts	2
4/13	Team Meeting/Adjusting tables and inserts	2.5
4/14	Work on SQL tables and inserts	2
4/15	Run queries for M3 and HW5	4
4/16	Finalize M3 and HW5 submission requirements during Team Meeting	2
4/17	Create Access form for Customer table HW5	1
4/18	Team Meeting	2
4/22	Work on M4 Tasks (Views/Reports/Forms/Transactions)	4
4/23	Team Meeting/Finalize Milestone 4	7
4/25	Updating Data Dictionary to finalize for paper/presentation	3
4/27	Updating Data Dictionary to finalize for paper/presentation	2

4/28	Add ACL tests to final paper/presentation	5
	<b>Total Hours for Elijah Jackson:</b>	52.58

Group Member:	<b>Erika Moss</b>	
Date of Work Performed	Description of Work Performed	Time Spent on Work Performed (HRS)
3/21/2022	Introduction group meeting/Team Charter/Code of Ethics	1
3/23/2022	Team Meeting/Intro paragraph/ resume	1.75
3/26/2022	Team Meeting	0.25
3/28/2022	Group meeting/ER Diagram	2.5
3/30/2022	ERD	0.5
4/1/2022	ERD/Data Dictionary	0.5
4/2/2022	ERD	1.25
4/4/2022	Team Meeting	0.5
4/6/2022	ERD	0.75
4/7/2022	ERD	0.25
4/9/2022	ERD	0.25

Group Member:	<b>Erika Moss</b>	
4/11/2022	ERD/Database Data	2
4/11/2022	Team Meeting	2.75
4/13/22	Database	1.5
4/16/22	ERD/Queries/HW 5 and 6/Team Meeting	3.75
4/18/2022	ERD/Team meeting	2
4/21/2022	M4 Assignment	3
4/22/2022	Adding more data to database	0.5
4/23/2022	M4 Assignment/Team Meeting	5.25
4/24/2022	Team Meeting	1.75
4/26/2022	Final paper: organizing all versions of ERD, updating ERD, matching SQL to ERD	2.25
4/27/2022	Final paper: ERDs in final paper, writing client intro and situation requirements	1.75
4/27/2022	Team Meeting	.50
4/28/2022	Final paper: finished writing tasks assigned to me, worked on presentation slides	3.50
4/29/2022	Final presentation recording, worked on queries for final paper, updated timesheet, helped format paper	2

	<b>Total Hours for Erika Moss:</b>	<b>42 hrs</b>
--	------------------------------------	---------------

Team Member Name:	<b>Jorge Castro</b>	
Date of Work Performed	Description of Work Performed	Time Spent on Work Performed (HRS)
3/23/2022	Zoom Meeting/Milestone 1	0.5
3/24/2022	Resume/Paragraph	0.75
3/26/2022	Zoom Office Meeting	0.25
3/28/2022	Zoom Meeting/Milestone 2	2
4/2/2022	Zoom Meeting/Milestone 2	2

4/4/2022	Zoom Meeting/Milestone 3	1.5
4/9/2022	Zoom Meeting/Milestone 3	2
4/11/2022	Zooom Meeting/Milestone 3	4.5
4/12/2022	Zoom Meeting/Milestone 3	3
4/14/2022	Milestone 3	1
4/16/2022	Zoom Meeting/Milestone 3/HW5&6	4
Team Member Name:	<b>Jorge Castro</b>	
4/17/2022	Milestone 3 Queries / Complete Milestone PDF	4.5
4/18/2022	Team Zoom Meeting Assigning and Reviewing Milestone 4	2.5
4/23/2022	Team Zoom Meeting Milestone 4/ Triggers and Procedures	4
4/24/2022	Finalizing Report and Views	2.5
4/27/2022	Zoom meeting	.5
4/28/2022	Milestone 5+6	4
4/28/200	Milestone 5+6	5

Totals:	Totals for Jorge Castro	44.5
---------	-------------------------	------

<b>Ryu Lewis</b>		
Date of Work Performed	Description of Work Performed	Time Spent on Work Performed (HRS)
3/24	Team Charter/Code of Ethics	0.1
3/24	Introductory Paragraph	0.5
3/24	Resume	0.5
3/26	Meeting / Finalized Milestone 1	0.2
3/28	Meeting / Worked on Milestone 2	2
4/1	Reviewed Data Dictionary	0.5
4/2	Meeting / Finalized Milestone 2	1
4/4	Meeting / Wrote scripts	0.5
4/8	Updated ERD dictionary	1
4/9	Meeting / Worked on tables and values on Excel and SQL Server	3

4/10	Updated values for Orders and Invoice	1.5
4/11	Meeting / Added more values in Orders and Invoice	2
4/12	Meeting / Worked on fixing issues from the previous day	2
4/13	Worked on tables and the foreign keys / foreign key constraints	5
4/15	Worked on fixing issues in SQL Server / Worked on M3 queries	7
4/16	Meeting / Worked on Milestone 3 and Homeworks 5&6	3
4/18	Meeting	2
4/22	Worked on Milestone 4	4
4/23	Meeting / Finishing touches on Milestone 4	4
4/25	Meeting	2
4/27	Worked on M5 and M6	4

4/28	Continued Working on M5 and M6	4
	<b>Total</b>	49.8

**APPENDICES C**

**TEAM CHARTER AND MEMBERSHIP  
AGREEMENT &**

**TEAM MEETING AGENDAS**

We, the Beta Testers, have signed and agreed to the attached Team Membership Agreement. This Team Charter/Code of Ethics details the minimum expectations of appropriate behavior for each team member, and will be used to evaluate any potential problems.

In the future, we may decide that the rules and expectations detailed in this document need to be updated or changed. At that time, we would create a revised Charter/Code of Ethics. The revised version will then become effective, and the older version will become void.

### **Code of Conduct:**

- As a team, we will respect our team members in all forms of communication: group chats, emails, video conferences, and collaboration tools.
- We will support each other throughout the project and do our fair share of work.
- We will respect each other's responsibilities outside of school and make our best effort to be available and accountable during this project.

### **Participation:**

- We will each contribute to every project milestone and submission.
- We will review the work that has been completed to ensure project requirements are met and documents are professional.
- All Team Members are expected to complete their duties by the Team's agreed upon Saturday meeting time.
- Team members who do not complete their portion of the project before Saturday understand that they have hindered the continuation and completion of the project and may be subject to consequences eliminating them from the team.

### **Communication:**

- We will respond to all communications directed to us within 24 hours.
- We will give our team appropriate notice if we are unable to attend a meeting or complete a portion of a project requirement.

### **Problem Solving:**

- We will collaborate to reach a decision together.
- We will give the team members involved in the conflict an opportunity to share their views of the situation.

### **Meeting Guidelines:**

- Group Meetings will be held via Zoom on Monday and Saturday evenings (from 7pm-10pm MST). Mondays will be used to discuss expectations of duties for the coming

week and how they apply to the project. Saturdays will be used for finalizing documentation that will be submitted.

- We will arrive at meetings on time, both in person or over video conferencing.
- We will respect the ideas and opinions of others by listening to them fully without interrupting.
- We will proceed with agenda items in a timely manner, keeping the schedules of our teammates in mind and respecting their time.

We, the members of Beta Testers, accept the conditions of team membership for CIS 360. The specific conditions of team membership are recorded in the Team Charter which we have developed above.

The Charter reflects the minimum level of performance that is expected of any group member. If a group member fails to meet this minimum level, they may be subject to consequences as follows:

1. The first time a group member fails to meet their responsibilities, the group will discuss the actions in a team meeting or email addressed to the offender from the group.
2. The second infraction by a team member will result in a group meeting and a reminder that further infractions will result in an email to the professor.
3. If the problem continues, the group will discuss the situation with the professor and ask that they be removed from the team.

Signing this document certifies that each team member agrees to comply with all aspects of the Team Charter, and understands that the consequences of failing to meet their team responsibilities include the final possibility of being removed from the group and having to complete the project alone.

**Project Team Name: Beta Testers**

Typed Name:	Signature	Date:
Alec Valencia	<i>Alec Valencia</i>	03/22/2022
Erika Moss	<i>Erika Moss</i>	03/23/2022
Celeste Garcia	<i>Celeste Garcia</i>	03/24/2022
Christine Beck	<i>Christine Beck</i>	03/23/2022
Elijah Jackson	<i>Elijah Jackson</i>	03/23/2022

Ryu Lewis	<i>Ryu Lewis</i>	03/24/2022
Jorge Castro	<i>Jorge Castro</i>	03/25/2022

## TEAM MEETING AGENDAS

3/21 Team meeting 7pm MST

- Participants: Erika and Christine
- First meeting, planning the next full group meeting.

3/23 Team meeting 7pm MST

- Participants: Jorge, Alec, Celeste, Christine, Erika
- Delegation of duties for M1

3/26 Team Meeting 7pm MST

- Participants: Alec, Celeste, Christine, Erika, Elijah, Ryu, Jorge
- Wrap up M1

3/28 Team meeting 7pm MST

- Participants: Alec, Celeste, Christine, Erika, Elijah, Ryu, Jorge
- Delegation of duties for M2

4/02: Team meeting 7pm MST

- Participants: Alec, Celeste, Christine, Erika, Elijah, Ryu, Jorge
- Collaboration and finalization M2

4/04 Team meeting 7pm MST

- Participants: Alec, Celeste, Christine, Erika, Elijah, Ryu, Jorge
- Delegation of duties for M3

4/11 Team meeting 7pm MST

- Participants: Alec, Celeste, Christine, Erika, Elijah, Ryu, Jorge
- Collaborative work on script inserting for M3

4/12 Team meeting 7pm MST

- Participants: Alec, Celeste, Elijah, Ryu, Jorge
- Collaboration on revisioning tables and insert for M3.

4/16 Team meeting 7pm MST

- Participants: Alec, Celeste, Christine, Erika, Elijah, Ryu, Jorge
- Collaboration on M4 Queries, Access, ERD comparison and HW 5 & 6 completion.

4/17 or 4/18 Team meeting 7pm MST

- Participants: Alec, Celeste, Christine, Erika, Elijah, Ryu, Jorge
- Collaboration on completion of tables and re-inserts, assigned queries and screen shots for M3

4/23 Team meeting 7pm MST

- Participants: Alec, Celeste, Christine, Erika, Elijah, Ryu, Jorge
- Wrap up M4

4/25 Team meeting 7pm MST

- Participants: Alec, Celeste, Christine, Erika, Elijah, Ryu, Jorge
- Delegation of final duties

4/26 Team meeting 7pm MST

- Participants: Christine and Celeste
- Discussed current status of progress on Final duties

4/28 Team meeting 7pm MST

- Participants: Celeste, Erika, Christine, Jorge, Ryu and Alec
- Discussed status of final project

4/29 Team Meeting 7pm MST

- Participants: Celeste, Alec
- To Discuss any questions or updates regarding final duties

4/30 Final Team Meeting 7pm MST

- Participants: Ryu and Celeste
- Final wrap up of project completion.