**ITEC 4200 Advanced Database Semester Project**

Project Name: Concert Halls

Spring 2021

Samuel Mckinney & Yaris Cotto (Group 3)

[smckinney1@ggc.edu](mailto:smckinney1@ggc.edu) & [ycottoquinones@ggc.edu](mailto:ycottoquinones@ggc.edu)

Table of Contents

[PART I. Executive Summary 4](#_Toc70454350)

[Objective 4](#_Toc70454351)

[Goals 4](#_Toc70454352)

[Solution 4](#_Toc70454353)

[Benefits 4](#_Toc70454354)

[Project Outline 4](#_Toc70454355)

[Schema Design 5](#_Toc70454356)

[Entity-Relationship Diagram 6](#_Toc70454357)

[PART II. DDL: Schema Implementation 7](#_Toc70454358)

[PART III. DML: Table Implementation 8](#_Toc70454359)

[Customer Outputs and Table 8](#_Toc70454360)

[OrderT Outputs and Table 10](#_Toc70454361)

[Hall Outputs and Table 12](#_Toc70454362)

[Performer Outputs and Table 14](#_Toc70454363)

[PerformanceT Outputs and Table 16](#_Toc70454364)

[Tickets Outputs and Table 18](#_Toc70454365)

[Part IV. Queries 20](#_Toc70454366)

[Queries 1-3 20](#_Toc70454367)

[Query 1 20](#_Toc70454368)

[Query 2 20](#_Toc70454369)

[Query 3 21](#_Toc70454370)

[Queries 4-5 21](#_Toc70454371)

[Query 4 21](#_Toc70454372)

[Query 5 22](#_Toc70454373)

[Queries 6-7 22](#_Toc70454374)

[Query 6 22](#_Toc70454375)

[Query 7 23](#_Toc70454376)

[Queries 8-9 23](#_Toc70454377)

[Query 8 23](#_Toc70454378)

[Query 9 24](#_Toc70454379)

[Queries 10-11 24](#_Toc70454380)

[Query 10 24](#_Toc70454381)

[Query 11 25](#_Toc70454382)

[Queries 12-13 25](#_Toc70454383)

[Query 12 25](#_Toc70454384)

[Query 13 26](#_Toc70454385)

[Queries 14-15 27](#_Toc70454386)

[Query 14 27](#_Toc70454387)

[Query 15 28](#_Toc70454388)

[PART V. Schema Objects 29](#_Toc70454389)

[View 1 29](#_Toc70454390)

[View 2 29](#_Toc70454391)

[Index Creation 30](#_Toc70454392)

[Flashback 31](#_Toc70454393)

# PART I. Executive Summary

### Objective

The purpose of this database is to help administrate the operations of multiple concert halls by sorting and ordering tickets, tracking available and booked concert halls, keeping track on customers, and sorting the performances being performed.

### Goals

This database aims to simplify the storage of data involved in the day-to-day operation of multiple concert halls by simply storing data and making use of queries and reports that would give reliable, relevant information to ensure smooth running, that no concert hall is overbooked, every performer can perform their booked performance, and that every paying customer gets a working ticket.

### Solution

We will be making a total of six tables using Oracle 18c Express to tackle the goals above: ***Customer, Hall, Order, PerformanceT, Ticket,*** and ***Performer***. All six tables would, at a bare minimum, be crucial to the daily operating of multiple concert halls and needed to properly schedule, run a ticketing system, and generate queries and reports needed for success.

### Benefits

A database will provide the means necessary to generate helpful information such as reports, trends, customer insights, or needed queries, but also help make certain information convenient and readily available to customers such as the status of their own ticket.

### Project Outline

This project will contain the following major components:

* Schema Design
* Entity-Relationship Diagram
* Table Implementation
* Queries
* Reports

## Schema Design

**CUSTOMER** (Customer\_ID, Last\_Name, First\_Name, Email\_Address)

**ORDER** (Order\_ID, Order\_Date, Total, Customer\_ID)

FK Customer\_ID -> CUSTOMER

**HALL** (Hall\_ID, CapacityH, Hall\_Name)

**PERFORMER** (Performer\_ID, Performer\_First\_Name, Performer\_Last\_Name, Genre)

**PERFORMANCET** (Performance\_ID, Hall\_ID, Performer\_ID, Performance\_Date)

FK Hall\_ID -> HALL

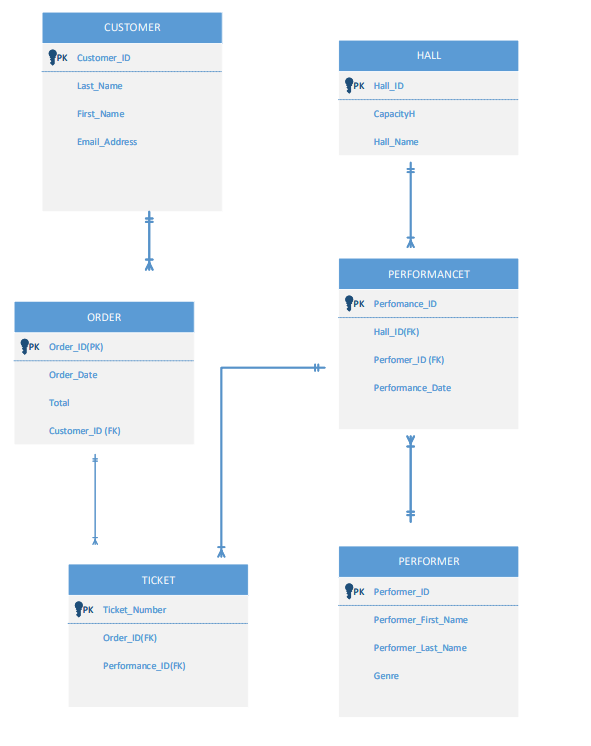
FK Performer\_ID -> PERFORMER

**TICKET** (Ticket\_Number, Order\_ID, Performance\_ID)

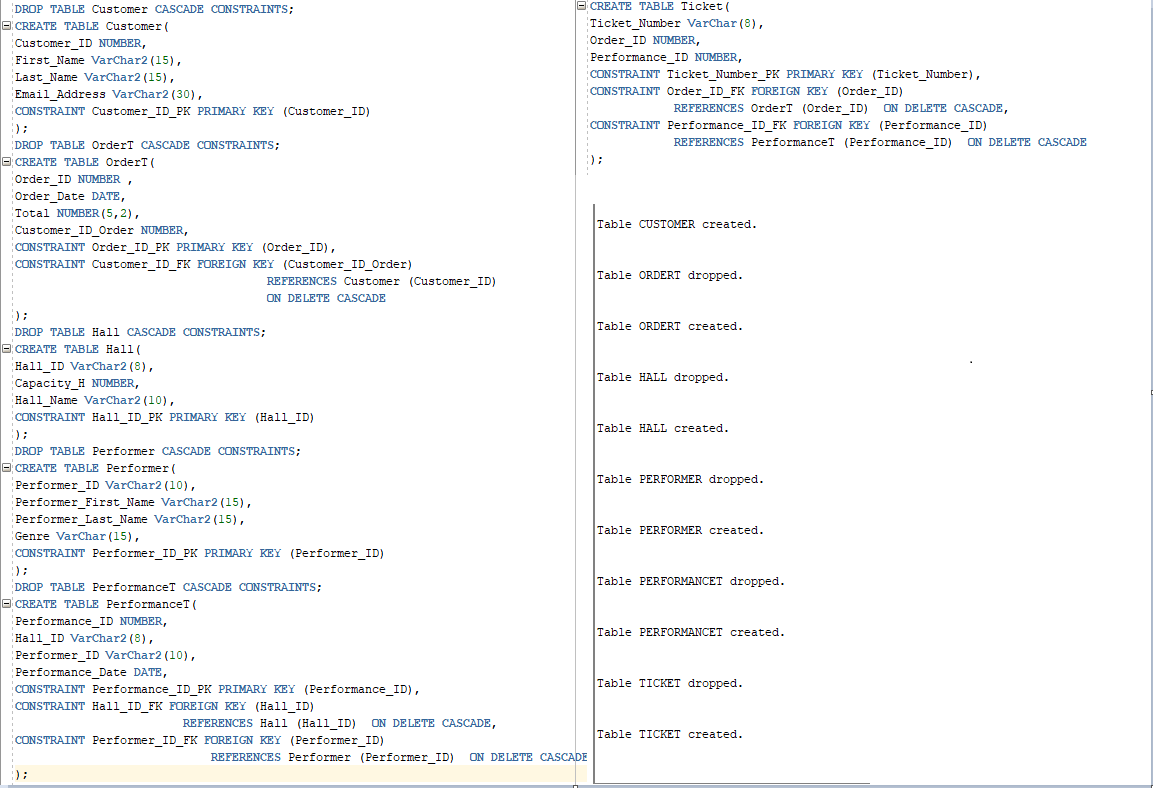
FK Order\_ID -> ORDER

FK Performance\_ID -> PERFORMANCET

## Entity-Relationship Diagram



# PART II. DDL: Schema Implementation

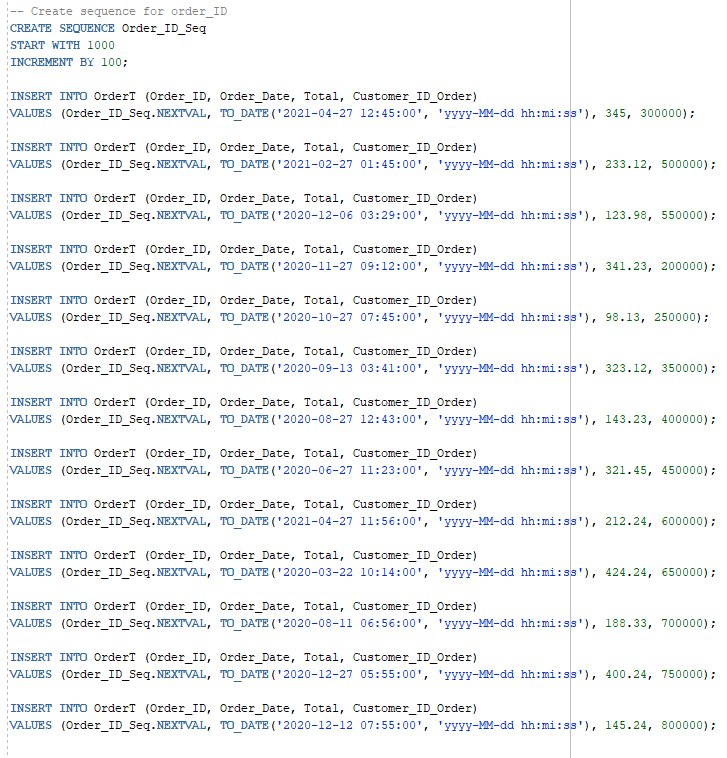


# PART III. DML: Table Implementation

## Customer Outputs and Table



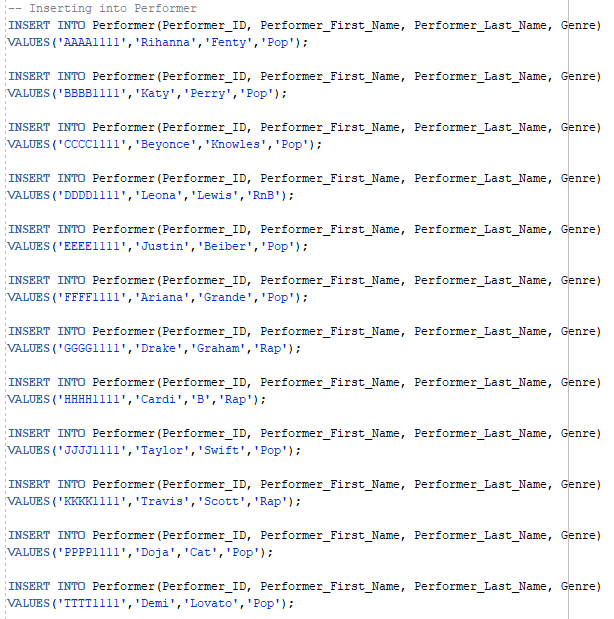
## OrderT Outputs and Table



## Hall Outputs and Table



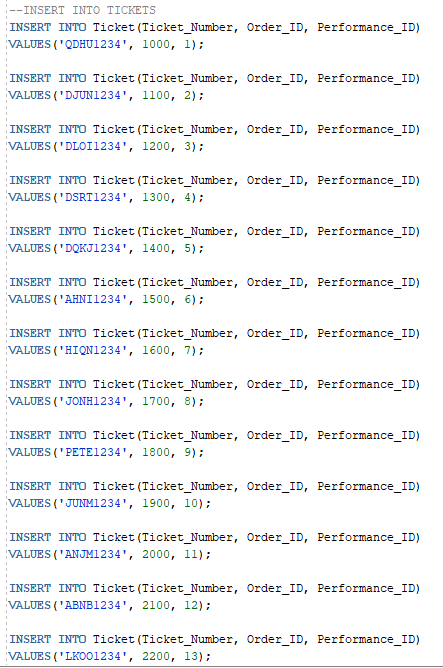
## Performer Outputs and Table



## PerformanceT Outputs and Table



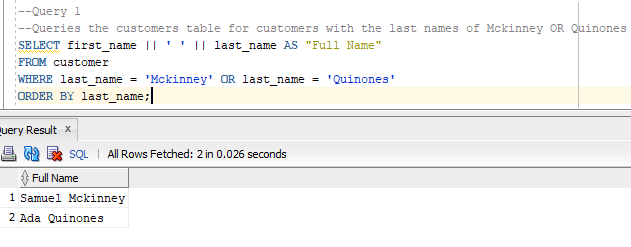
## Tickets Outputs and Table



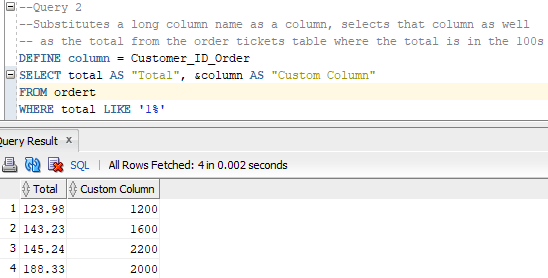
# Part IV. Queries

## Queries 1-3

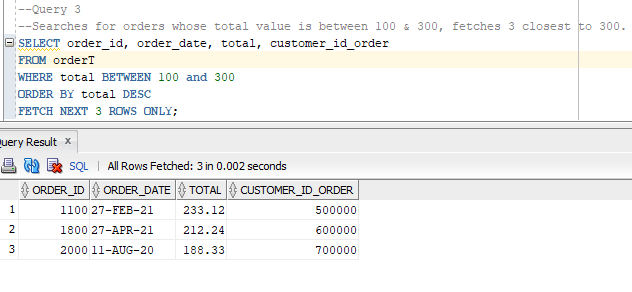
### Query 1



### Query 2

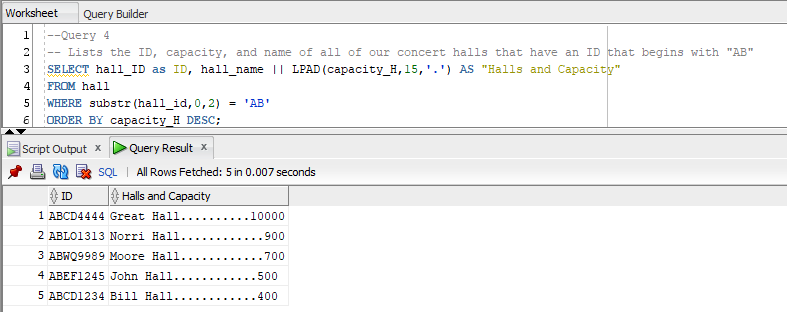


### Query 3

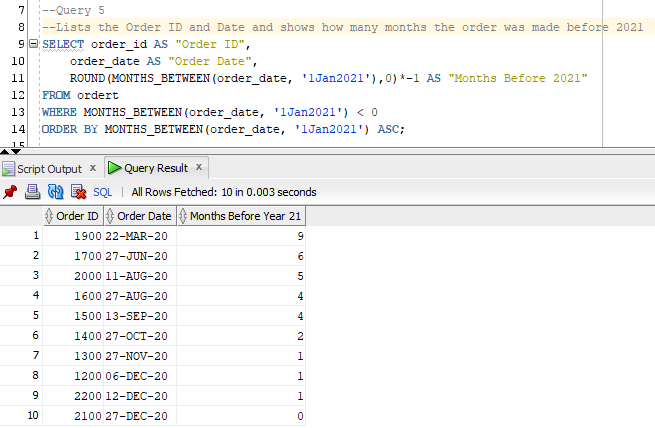


## Queries 4-5

### Query 4

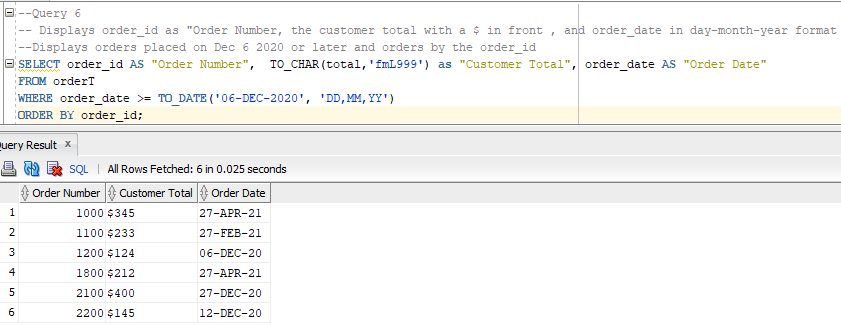


### Query 5

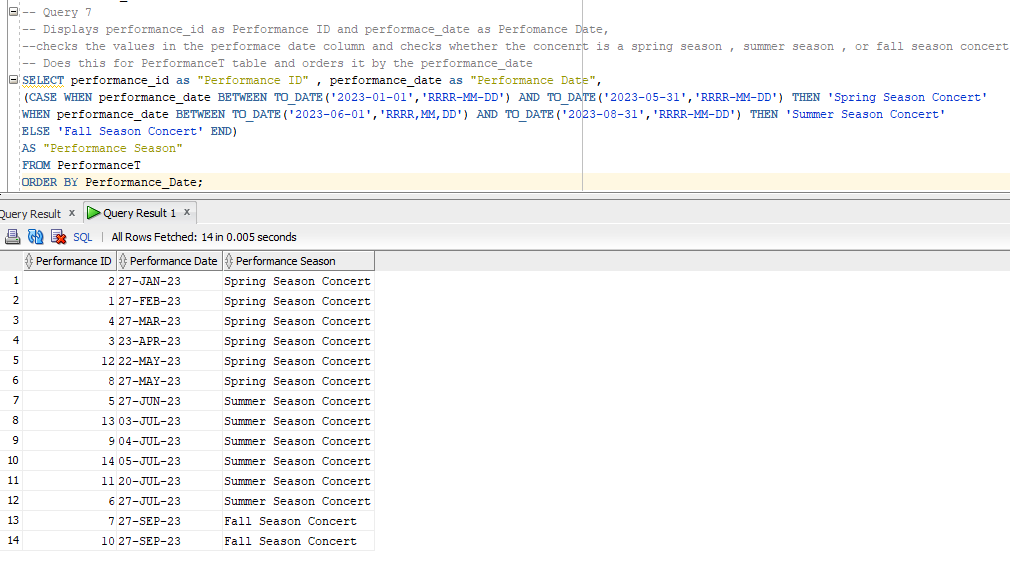


## Queries 6-7

### Query 6

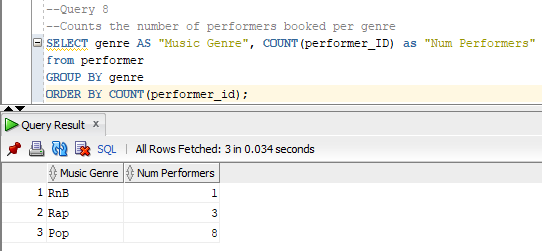


### Query 7

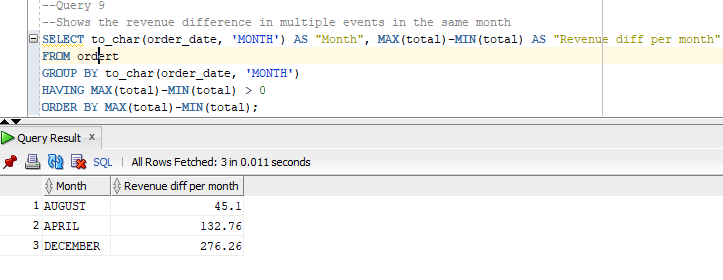


## Queries 8-9

### Query 8

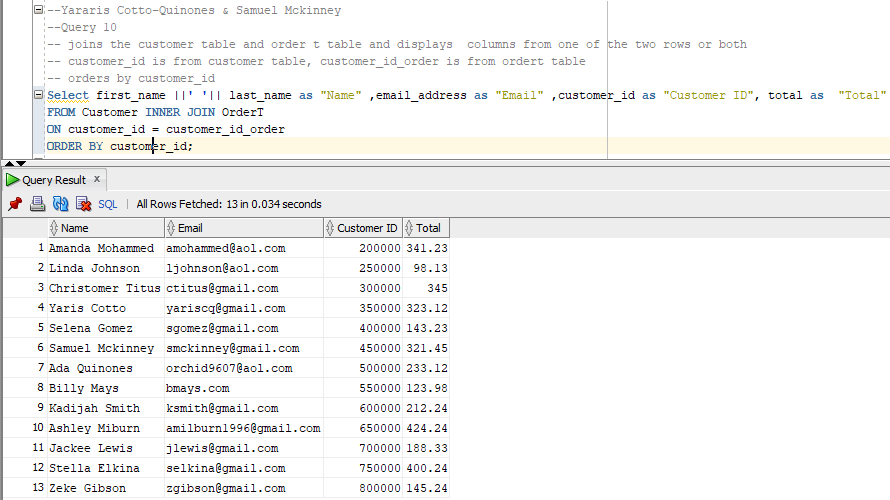


### Query 9

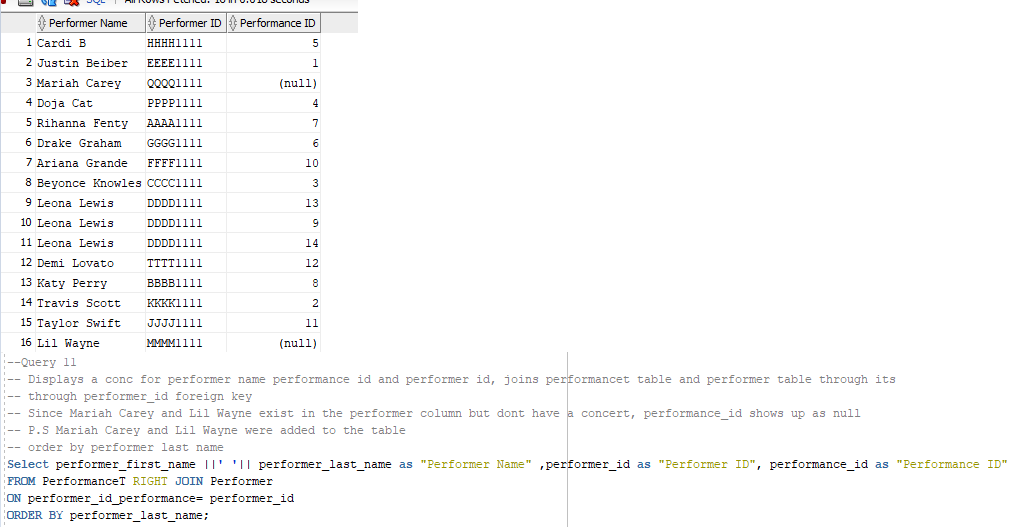


## Queries 10-11

### Query 10

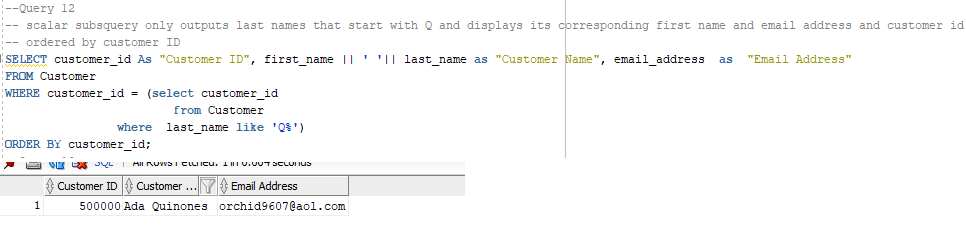


### Query 11

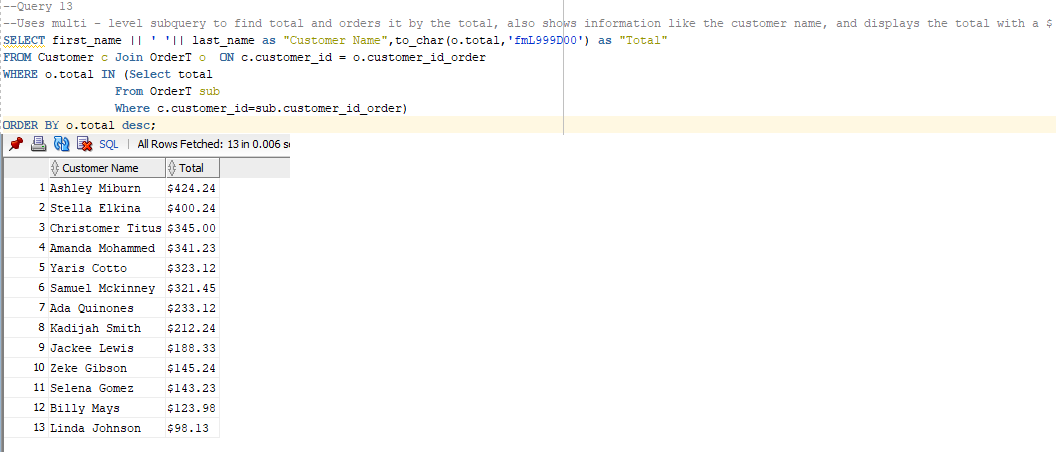


## Queries 12-13

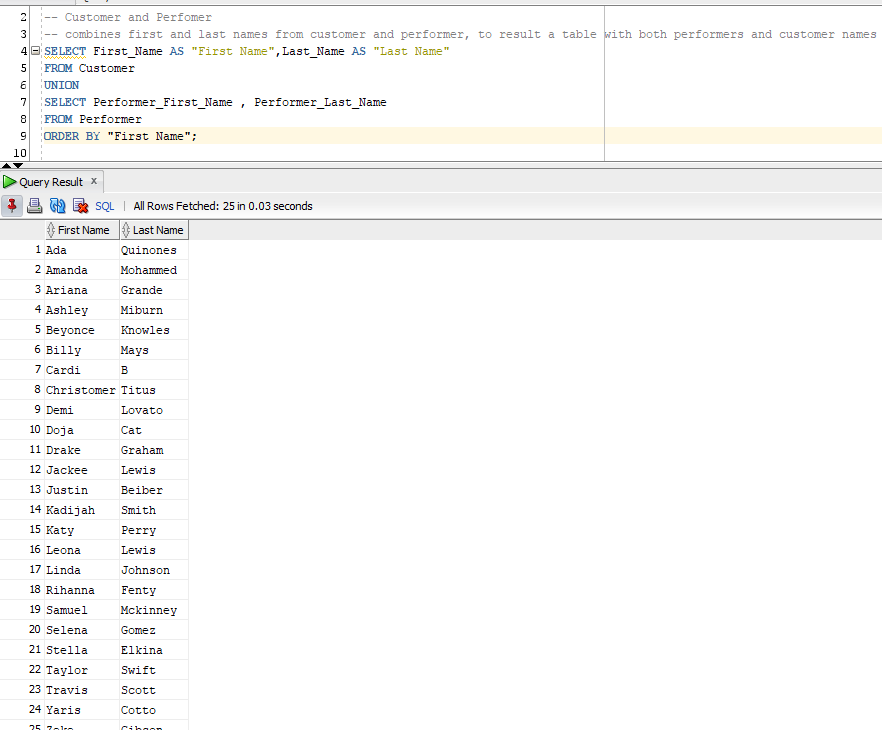
### Query 12



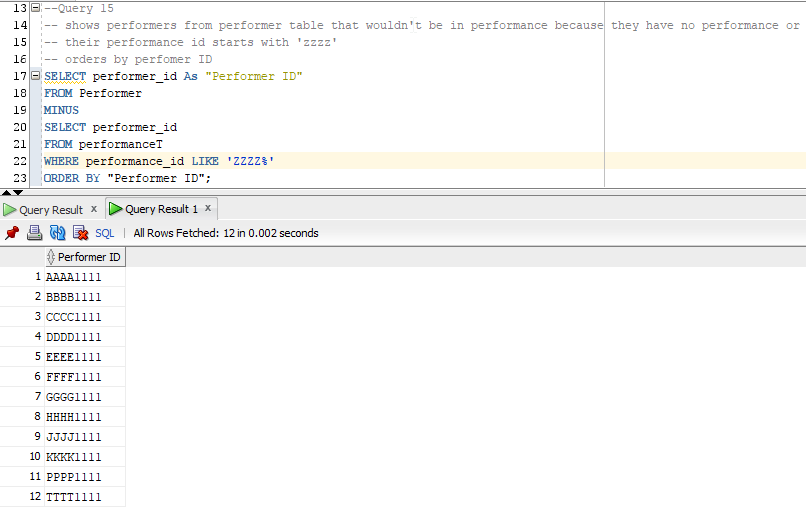
### Query 13



## Queries 14-15

Query 14

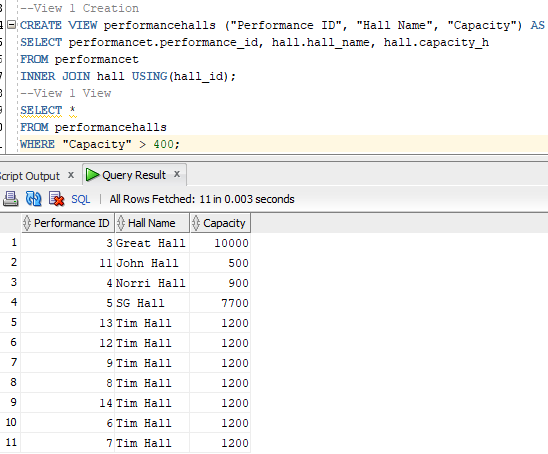
### Query 15



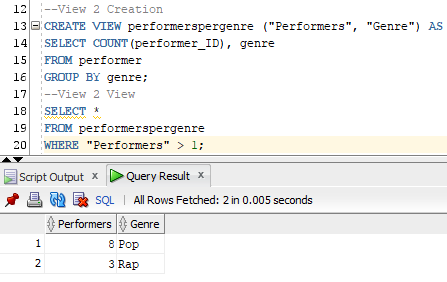
# 

# PART V. Schema Objects

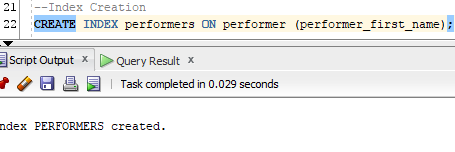
### View 1



### View 2



### Index Creation



### Flashback

