Ozzapalooza

Database Design Document

Andrew Baran

CMPT 308

Marist College

**Table of Contents:**

**Overview:**

**Executive Summary:**

**Overview:**

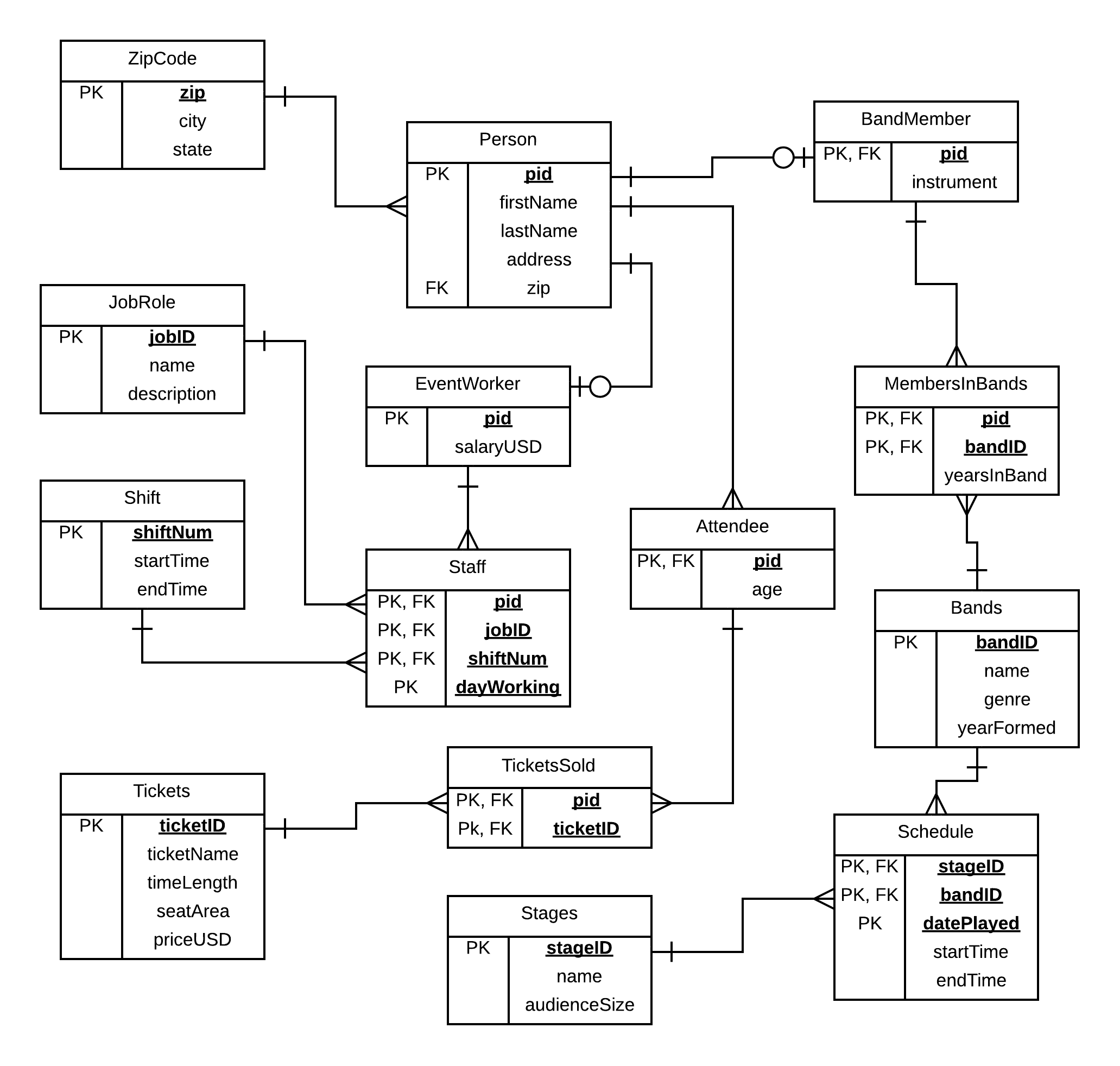
Ozzapalooza is one of the premiere music festivals in the United States. Located in and around the area of Phoenix, Arizona, this three-day long festival of music, energy, and fun brings in tens of thousands of individuals from all over the country. Many genres of music are heard, from classic rock to death metal, and many classic bands, like Rush and Slayer, perform each year to their fans.

**Objectives:**

Our consulting company has been hired by the organizers of the Ozzapalooza to create an appropriate database system for them to keep better track of the information regarding their employees, bands, and the attendees of the festival. Our goal was to create this system so that it would be very easy to modify the information necessary to the festival, such as the scheduling the bands and employees. As well as creating the necessary structure for the database, we have also provided the organizers of the festival with a variety of stored procedures and triggers to ensure the reliability of the system and the data within it, as well as queries to help in creating the necessary reports and statistics of the festival.

The outline of our proposed Database system is within this document. Each page contains a separate entity of the database system, which in total makes up the schema of the database system. As well as each entity and its functional dependencies being shown, the views, stored procedures, triggers, and security permissions for the database schema are also shown. This database system was designed in PostgreSQL 9.3.2, and it has been tested thoroughly to ensure accuracy and reliability.

**ER Diagram:**



**Tables**

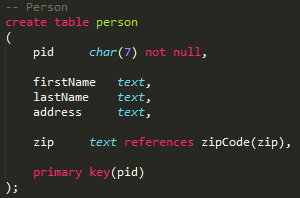
**Person Table**

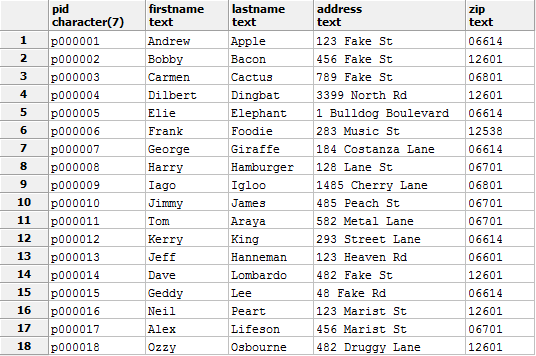
Purpose:

Table that stores the information about each person involved in the festival (attendees, event worker, and band members). Identifying information about the individual is stored in this table.

Functional Dependencies:

pid -> firstName, lastName, address, zip





**Tables**

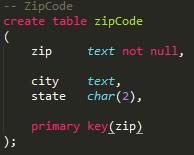
**ZipCode Table**

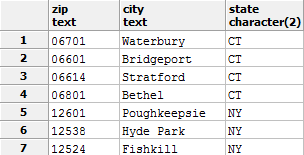
Purpose:

Table that stores the information of the zip codes of each person involved in the festival. Each zip code is associated with its corresponding city and state.

Functional Dependencies:

zip -> city, state





**Tables**

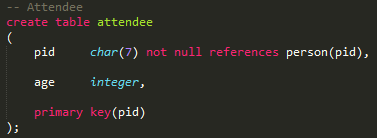
**Attendee Table**

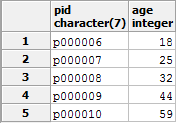
Purpose:

Table that stores the information of each person (customer) attending the festival, along with their age. This table is an entity subtype of the table Person.

Functional Dependencies:

pid -> age





**Tables**

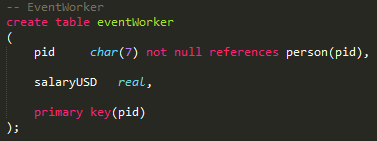
**EventWorker Table**

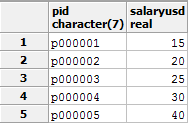
Purpose:

Table that stores the information about each employee that works at the festival, along with that employee’s salary in USD. This table is an entity subtype of the table Person

Functional Dependencies:

pid -> salaryUSD





**Tables**

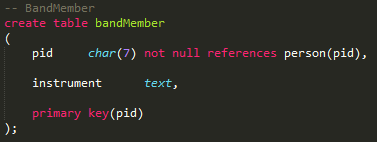
**BandMember Table**

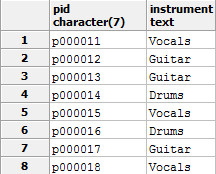
Purpose:

Table that stores the information about each band member, which includes which instrument they play. This table is an entity subtype of the Person table.

Functional Dependencies:

pid -> instrument





**Tables**

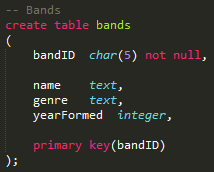
**Bands Table**

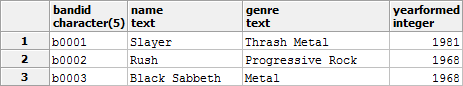
Purpose:

Table that stores the information of each band, including their name, genre, and other information about their history.

Functional Dependencies:

bandID -> name, genre, yearFormed





**Tables**

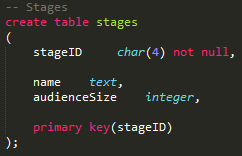
**Stages Table**

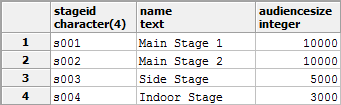
Purpose:

Table that stores the information about each stage that is at the festival, which includes the capacity of each audience.

Functional Dependencies:

stageID -> name, audienceSize





**Tables**

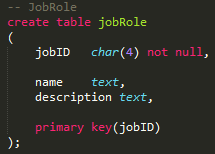
**JobRole Table**

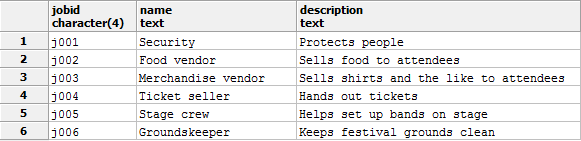
Purpose:

Table that stores the information about each job that an event worker at the festival can perform, including a description of that job

Functional Dependencies:

jobID -> name, description





**Tables**

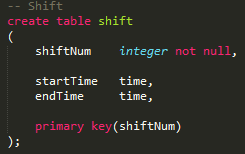
**Shift Table**

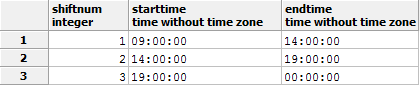
Purpose:

Table that stores the information about each shift that an event worker can work during, including the time frame of each shift.

Functional Dependencies:

shiftNum -> startTime, endTime





**Tables**

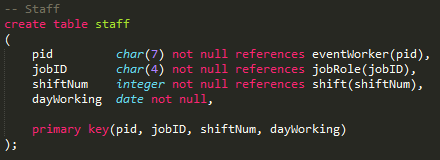
**Staff Table**

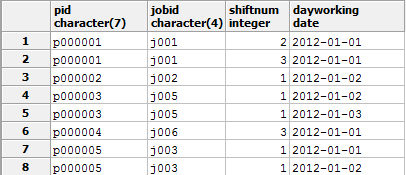
Purpose:

Table that stores the information about each employee, when they are working and what job they are working on for the given day.

Functional Dependencies:

pid + jobID + shiftNum + daysWorking ->





**Tables**

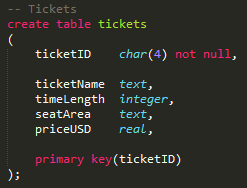
**Tickets Table**

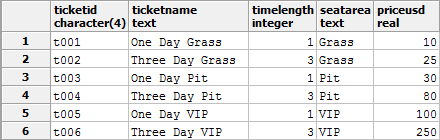
Purpose:

Table that stores the information about each type of ticket that is available for sale.

Functional Dependencies:

ticketID -> ticketName, timeLength, seatArea, priceUSD





**Tables**

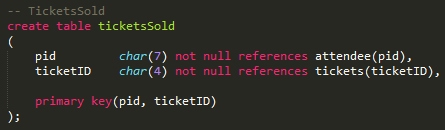
**TicketsSold Table**

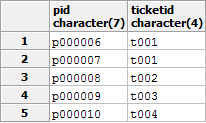
Purpose:

Table that stores the information about which attendee bought which ticket for the festival.

Functional Dependencies:

pid + ticketID ->





**Tables**

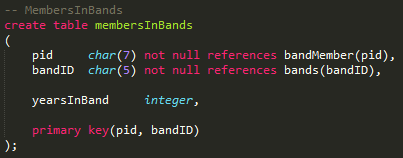
**MembersInBands Table**

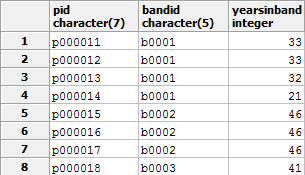
Purpose:

Table that stores the information about which band members are in which band, as well as how long they have been in that band.

Functional Dependencies:

pid + bandID -> yearsInBand





**Tables**

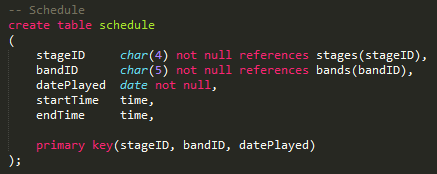
**Schedule Table**

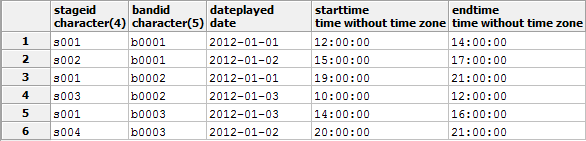
Purpose:

Table that stores the information about which band plays on which stage at a given day, as well as the start and end time of that bands performance.

Functional Dependencies:

stageID + bandID + datePlayed -> startTime, endTime





**Views**

**Band Information View**

Purpose:

View that allows a person to see a succinct summary of a band, its members, and other important information.

create view bandInformation as

select person.firstName, person.lastName, bands.name as bandName, bandMember.instrument, membersInBands.yearsInBand

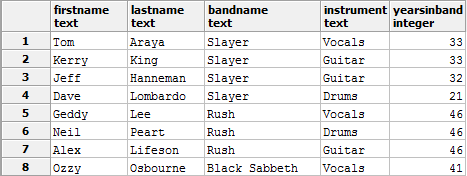
from bands, bandMember, membersInBands, person

where bands.bandID = membersInBands.bandID

and bandMember.pid = membersInBands.pid

and person.pid = bandMember.pid;

Results:



**Views**

**Complete Schedule View**

Purpose:

View that allows a person to see the complete schedule of the festival, including which band is on which stage at a certain time.

create view completeSchedule as

select stages.name as stageName, bands.name as bandName, schedule.datePlayed, schedule.startTime, schedule.endTime

from schedule, bands, stages

where schedule.bandID = bands.bandID

and schedule.stageID = stages.stageID;

Results:

