```
-- phpMyAdmin SQL Dump
-- version 4.9.0.1
-- https://www.phpmyadmin.net/
--
-- Host: localhost:8889
-- Generation Time: Jan 19, 2020 at 07:02 AM
-- Server version: 5.7.26
-- PHP Version: 7.3.7

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `DATABASE PROJECT`
--
CREATE DATABASE IF NOT EXISTS `DATABASE PROJECT` DEFAULT CHARACTER SET
utf8 COLLATE utf8_general_ci;
USE `DATABASE PROJECT`;

-- --------------------------------------------------------


--
-- Table structure for table `Address`
--

DROP TABLE IF EXISTS `Address`;
CREATE TABLE `Address` (
  `Employee_EmpID` text NOT NULL,
  `Customer_ID` text NOT NULL,
  `Street` text NOT NULL,
  `City` text NOT NULL,
  `State` text NOT NULL,
  `Zip` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `Address`
--

INSERT INTO `Address` (`Employee_EmpID`, `Customer_ID`, `Street`,
`City`, `State`, `Zip`) VALUES
('111', 'NULL', '4 Sam St', 'Blacksburg', 'VA', 24060),
('112', 'NULL', '5 Sam St', 'Blacksburg', 'VA', 24060),
('111', '1111', '25 Back Rd', 'Blacksburg', 'VA', 24060),
```

```
('113', 'NULL', '1738 Yeet Dr.', 'Christiansburg', 'VA', 24073),
('114', 'NULL', '9 Center St.', 'Blacksburg', 'VA', 24060),
('115', 'NULL', '710 Harding Ave.', 'Blacksburg', 'VA', 24060),
('116', 'NULL', '11 Pheasant Run Dr.', 'Blacksburg', 'VA', 24060),
('117', 'NULL', '99 Barstool St.', 'Christiansburg', 'VA', 24073),
('118', 'NULL', '14 Beamer Dr.', 'Blacksburg', 'VA', 24060),
('119', 'NULL', '16 Beamer Dr.', 'Blacksburg', 'VA', 24060),
('120', 'NULL', '24 Center St.', 'Blacksburg', 'VA', 24060),
('111', '1111', '25 Back Rd.', 'Blacksburg', 'VA', 24060),
('111', '1112', '17 Main St.', 'Blacksburg', 'VA', 24060),
('111', '1113', '42 Main St', 'Blacksburg', 'VA', 24060),
('111', '1114', '7 Windy St.', 'Chester Springs', 'PA', 19425),
('111', '1115', '29 Dirt Rd.', 'Blacksburg', 'VA', 24060),
('111', '1116', '33 Roanoke Ln.', 'Blacksburg', 'VA', 24060),
('111', '1117', '22 Lake Rd.', 'Chester Springs', 'PA', 19425),
('111', '1118', '21 Ocean Dr.', 'Christiansburg', 'VA', 24073),
('111', '1119', '98 Tremont Ln.', 'Downingtown', ' PA', 19341),
('111', '1120', '42 Hollywood Dr.', 'Blacksburg', 'VA', 24060);

-- --------------------------------------------------------

--
-- Table structure for table `Concession`
--

DROP TABLE IF EXISTS `Concession`;
CREATE TABLE `Concession` (
  `Sales_SalesID` text NOT NULL,
  `Concession_ItemName` text NOT NULL,
  `Concession_QuantitySold` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `Concession`
--

INSERT INTO `Concession` (`Sales_SalesID`, `Concession_ItemName`,
`Concession_QuantitySold`) VALUES
('1002', 'Popcorn', 1),
('1005', 'Skittles', 1),
('1005', 'Starburst', 1),
('1005', 'Chocolate', 1),
('1007', 'Popcorn', 1),
('1007', 'Skittles', 2),
('1011', 'Popcorn', 2),
('1011', 'Chocolate', 1),
('1012', 'Popcorn', 3),
('1013', 'Popcorn', 1);

-- --------------------------------------------------------
```

```
--
-- Table structure for table `Customer`
--

DROP TABLE IF EXISTS `Customer`;
CREATE TABLE `Customer` (
  `Customer_ID` int(11) NOT NULL,
  `Customer_Name` text NOT NULL,
  `Customer_LastName` text NOT NULL,
  `Customer_Age` date NOT NULL,
  `Customer_Email` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `Customer`
--

INSERT INTO `Customer` (`Customer_ID`, `Customer_Name`,
`Customer_LastName`, `Customer_Age`, `Customer_Email`) VALUES
(1111, 'Jerry', 'Garcia', '1942-08-01', 'Jgarcia@gmail.com'),
(1112, 'John', 'Lewis', '1997-03-11', 'JLewis@gmail.com'),
(1113, 'Ryan', 'Smith', '1998-09-21', 'RSmith@gmail.com'),
(1114, 'Jake', 'Watson', '1982-05-18', 'JWatson@gmail.com'),
(1115, 'Michael', 'Jordan', '1963-02-17', 'MJordan@gmail.com'),
(1116, 'Tiger', 'Woods', '1975-12-30', 'TWoods@gmail.com'),
(1117, 'Shaun', 'White', '1986-09-03', 'SWhite@gmail.com'),
(1118, 'Cheech', 'Chong', '1965-04-20', 'CheechnChong@gmail.com'),
(1119, 'Snoop', 'Dogg', '1971-10-20', 'Snoopy@gmail.com'),
(1120, 'Wiz', 'Khalifa', '1987-09-08', 'specialKK@gmail.com');

-- --------------------------------------------------------

--
-- Table structure for table `Employee`
--

DROP TABLE IF EXISTS `Employee`;
CREATE TABLE `Employee` (
  `Employee_EmpID` int(11) NOT NULL,
  `Employee_Name` text NOT NULL,
  `Employee_LastName` text NOT NULL,
  `Employee_Age` date NOT NULL,
  `Employee_PhoneNum` text NOT NULL,
  `Employee_DateHired` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `Employee`
--
```

```
INSERT INTO `Employee` (`Employee_EmpID`, `Employee_Name`,
`Employee_LastName`, `Employee_Age`, `Employee_PhoneNum`,
`Employee_DateHired`) VALUES
(111, 'Alex', 'Smith', '1995-06-01', '703-111-739', '2018-09-22'),
(112, 'Beck', 'Worth', '1998-10-22', '703-123-740', '2018-02-22'),
(113, 'Chris', 'Hemlock', '1993-03-11', '703-123-741', '2019-10-21'),
(114, 'Dave', 'Jones', '1997-10-30', '703-123-742', '2016-11-03'),
(115, 'Ellie', 'Swann', '2001-12-25', '703-123-743', '2018-04-23'),
(116, 'Frank', 'Beamer', '1999-11-03', '703-123-744', '2019-11-09'),
(117, 'Julia', 'Russo', '1994-04-20', '703-123-745', '2020-01-04'),
(118, 'Katie', 'Wilson', '1996-02-17', '703-123-746', '2019-04-02'),
(119, 'Hannah', 'Jane', '1992-09-26', '703-123-747', '2018-12-11'),
(120, 'Eli', 'Samuel', '1996-08-11', '703-123-748', '2019-07-16');

--
-- Triggers `Employee`
--
DROP TRIGGER IF EXISTS `updateAll_EmpID`;
DELIMITER $$
CREATE TRIGGER `updateAll_EmpID` AFTER UPDATE ON `Employee` FOR EACH
ROW BEGIN
/*
Updates employee id, when employee id is changed due to promotion.
*/
        UPDATE Address SET Address.Employee_EmpID = NEW.Employee_EmpID
    WHERE Address.Employee_EmpID = OLD.Employee_EmpID;

END
$$
DELIMITER ;

-- --------------------------------------------------------

--
-- Table structure for table `Location`
--

DROP TABLE IF EXISTS `Location`;
CREATE TABLE `Location` (
  `Location_TheaterID` int(11) NOT NULL,
  `Location_SeatsAvail` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `Location`
--

INSERT INTO `Location` (`Location_TheaterID`, `Location_SeatsAvail`)
VALUES
```

```
(1, 50),
(2, 50),
(3, 50),
(4, 50),
(5, 50);

-- --------------------------------------------------------

--
-- Table structure for table `Movie`
--

DROP TABLE IF EXISTS `Movie`;
CREATE TABLE `Movie` (
  `Movie_MovieID` int(11) NOT NULL,
  `Movie_Name` text NOT NULL,
  `Movie_Runtime` text NOT NULL,
  `Movie_Description` text NOT NULL,
  `Movie_Rating` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `Movie`
--

INSERT INTO `Movie` (`Movie_MovieID`, `Movie_Name`, `Movie_Runtime`,
`Movie_Description`, `Movie_Rating`) VALUES
(101, 'Star Wars', '1:32:15', 'Action in space', 'PG-13'),
(102, 'Space Jam', '1:48:52', 'Basketball with cartoons and michael
jordan', 'G'),
(103, 'Avatar', '2:12:15', 'Blue people get attacked by humans',
'PG-13'),
(104, 'Will Smith', '1:22:00', 'The life of Will Smith', 'R'),
(105, 'AVENGERS', '1:00:00', 'Thanos is in the last movie', 'PG-13');

--
-- Triggers `Movie`
--
DROP TRIGGER IF EXISTS `update_MovieID`;
DELIMITER $$
CREATE TRIGGER `update_MovieID` AFTER UPDATE ON `Movie` FOR EACH ROW
BEGIN
/*
Updates movie id because movie ids are reassigned so that the newer
movies have the higher values while older movies have lower values.
*/
UPDATE Showing SET Showing.Movie_MovieID = NEW.Movie_MovieID WHERE
Showing.Movie_MovieID = OLD.Movie_MovieID ;
END
$$
```

```
DELIMITER ;

-- ----------------------------------------------------------


--
-- Table structure for table `Sales`
--

DROP TABLE IF EXISTS `Sales`;
CREATE TABLE `Sales` (
  `Sales_SalesID` int(11) NOT NULL,
  `Employee_EmpID` int(11) NOT NULL,
  `Sales_ItemType` text NOT NULL,
  `Sales_Date` date NOT NULL,
  `Sales_Price` decimal(10,0) NOT NULL,
  `Sales_QuantitySold` int(11) NOT NULL,
  `Customer_ID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `Sales`
--

INSERT INTO `Sales` (`Sales_SalesID`, `Employee_EmpID`,
`Sales_ItemType`, `Sales_Date`, `Sales_Price`, `Sales_QuantitySold`,
`Customer_ID`) VALUES
(1001, 111, 'Ticket', '2019-11-22', '20', 2, 1113),
(1002, 113, 'Concession', '2019-11-22', '10', 1, 1113),
(1003, 111, 'Ticket', '2019-11-22', '35', 4, 1117),
(1004, 112, 'Ticket', '2019-11-22', '25', 3, 1114),
(1005, 113, 'Concession', '2019-11-22', '15', 3, 1114),
(1006, 111, 'Ticket', '2019-11-22', '50', 5, 1118),
(1007, 114, 'Concession', '2019-11-22', '20', 3, 1118),
(1008, 112, 'Ticket', '2019-11-22', '10', 1, 1111),
(1009, 115, 'Ticket', '2019-11-23', '20', 3, 1119),
(1010, 112, 'Ticket', '2019-11-23', '35', 4, 1112),
(1011, 113, 'Concession', '2019-11-23', '25', 3, 1112),
(1012, 114, 'Concession', '2019-11-23', '30', 3, 1111),
(1013, 113, 'Concession', '2019-11-23', '10', 1, 1119),
(1014, 112, 'Ticket', '2019-11-23', '20', 2, 1115),
(1015, 115, 'Ticket', '2019-11-23', '30', 4, 1113),
(1016, 112, 'Ticket', '2019-11-23', '10', 1, 1111);

-- ----------------------------------------------------------

--
-- Table structure for table `Showing`
--

DROP TABLE IF EXISTS `Showing`;
```

```
CREATE TABLE `Showing` (
  `Showing_ShowingID` int(11) NOT NULL,
  `Movie_MovieID` int(11) NOT NULL,
  `Location_TheaterID` int(11) NOT NULL,
  `Showing_StartTime` datetime NOT NULL,
  `Showing_EndTime` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `Showing`
--

INSERT INTO `Showing` (`Showing_ShowingID`, `Movie_MovieID`,
`Location_TheaterID`, `Showing_StartTime`, `Showing_EndTime`) VALUES
(984, 116, 1, '2019-11-23 08:00:00', '2019-11-23 09:22:00'),
(985, 115, 1, '2019-11-23 02:00:00', '2019-11-23 03:22:00'),
(986, 114, 3, '2019-11-23 08:00:00', '2019-11-23 10:12:15'),
(987, 113, 3, '2019-11-23 05:00:00', '2019-11-23 07:12:15'),
(988, 112, 2, '2019-11-23 06:00:00', '2019-11-23 07:48:52'),
(989, 111, 2, '2019-11-23 04:00:00', '2019-11-23 05:48:52'),
(990, 110, 1, '2019-11-23 06:00:00', '2019-11-23 07:32:15'),
(991, 109, 1, '2019-11-23 04:00:00', '2019-11-23 05:32:15'),
(992, 108, 1, '2019-11-22 08:00:00', '2019-11-22 09:22:00'),
(993, 107, 1, '2019-11-22 02:00:00', '2019-11-22 03:22:00'),
(994, 106, 3, '2019-11-22 08:00:00', '2019-11-22 10:12:15'),
(995, 105, 3, '2019-11-22 05:00:00', '2019-11-22 07:12:15'),
(996, 104, 2, '2019-11-22 06:00:00', '2019-11-22 07:48:52'),
(997, 103, 2, '2019-11-22 04:00:00', '2019-11-22 05:48:52'),
(998, 101, 1, '2019-11-22 06:00:00', '2019-11-22 07:32:15'),
(999, 101, 1, '2019-11-22 04:00:00', '2019-11-22 05:32:15');

-- --------------------------------------------------------

--
-- Table structure for table `Ticket`
--

DROP TABLE IF EXISTS `Ticket`;
CREATE TABLE `Ticket` (
  `Sales_SalesID` int(11) NOT NULL,
  `Movie_MovieID` int(3) NOT NULL,
  `Ticket_Type` text NOT NULL,
  `Ticket_SeatNum` int(11) NOT NULL,
  `Showing_ShowingID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `Ticket`
--
```

```
INSERT INTO `Ticket` (`Sales_SalesID`, `Movie_MovieID`, `Ticket_Type`,
`Ticket_SeatNum`, `Showing_ShowingID`) VALUES
(1001, 101, 'Adult', 1, 999),
(1001, 101, 'Adult', 2, 999),
(1003, 103, 'Adult', 1, 997),
(1003, 103, 'Adult', 2, 997),
(1003, 103, 'Adult', 3, 997),
(1004, 101, 'Adult', 3, 999),
(1004, 101, 'Child', 5, 999),
(1003, 103, 'Child', 4, 997),
(1004, 101, 'Adult', 4, 999),
(1006, 103, 'Adult', 5, 997),
(1006, 103, 'Adult', 6, 997),
(1006, 103, 'Adult', 7, 997),
(1006, 103, 'Adult', 8, 997),
(1006, 103, 'Adult', 9, 997),
(1006, 103, 'Adult', 10, 997),
(1008, 103, 'Adult', 10, 997),
(1009, 102, 'Adult', 1, 998),
(1009, 102, 'Child', 2, 998),
(1009, 102, 'Child', 3, 998),
(1010, 101, 'Adult', 1, 999),
(1010, 101, 'Adult', 2, 999),
(1010, 101, 'Adult', 3, 999),
(1010, 101, 'Child', 4, 999),
(1014, 102, 'Adult', 4, 998),
(1014, 102, 'Adult', 5, 998),
(1014, 102, 'Adult', 6, 998),
(1015, 101, 'Adult', 5, 999),
(1015, 101, 'Adult', 6, 999),
(1015, 101, 'Child', 7, 999),
(1015, 101, 'Child', 8, 999),
(1016, 102, 'Adult', 6, 998);

--
-- Indexes for dumped tables
--


--
-- Indexes for table `Customer`
--
ALTER TABLE `Customer`
  ADD PRIMARY KEY (`Customer_ID`);

--
-- Indexes for table `Employee`
--
ALTER TABLE `Employee`
  ADD PRIMARY KEY (`Employee_EmpID`);
```

```
--
-- Indexes for table `Location`
--
ALTER TABLE `Location`
  ADD PRIMARY KEY (`Location_TheaterID`);

--
-- Indexes for table `Movie`
--
ALTER TABLE `Movie`
  ADD PRIMARY KEY (`Movie_MovieID`);

--
-- Indexes for table `Sales`
--
ALTER TABLE `Sales`
  ADD PRIMARY KEY (`Sales_SalesID`);

--
-- Indexes for table `Showing`
--
ALTER TABLE `Showing`
  ADD PRIMARY KEY (`Showing_ShowingID`);

  /*
  PROCEDURES BELOW
  */

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `ShowingTimes`()
BEGIN
/*
Available showing times for all movies. This query allows employees to
see the showing time and location for all available movies.


*/

SELECT Movie.Movie_Name AS 'Movie Name', Showing.Showing_StartTime As
'Start Time', Showing.Showing_EndTime As 'End Time',
Showing.Location_TheaterID As 'Theater Number'
FROM Movie, Showing
WHERE Movie.Movie_MovieID = Showing.Movie_MovieID
ORDER BY Showing.Showing_StartTime;

END$$
DELIMITER ;

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `numConcession`()
```

```
BEGIN
/*
Number of sales for different concession items.
This query is important because it shows which concession items are
most popular and which are the least popular.

*/

    SELECT Concession.Concession_ItemName As 'Concession Item',
SUM( Concession.Concession_QuantitySold ) AS 'Amount Sold',
CONCAT('$', FORMAT(SUM(Sales.Sales_Price), 2)) AS MoneyMade FROM
Concession, Sales WHERE Concession.Sales_SalesID IN( SELECT
Sales.Sales_SalesID FROM Sales, Customer WHERE Sales.Customer_ID =
Customer.Customer_ID ) GROUP BY Concession.Concession_ItemName;
END$$
DELIMITER ;

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `concession_Sold`()
BEGIN
/*
Total dollar amount of sales for concession items.
This query is important because it'd show how much total money was
made on concession sales.
*/
SELECT Sales.Sales_ItemType 'Type of Item', CONCAT('$',
FORMAT(SUM(Sales.Sales_Price), 2)) AS Revenue FROM Sales WHERE
Sales.Sales_ItemType = 'Concession';



END$$
DELIMITER ;

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `empNumDays`()
BEGIN
/*
Number of days an employee has been with the company.
This will help management see if an employee is eligible for a
promotion.




*/

SELECT DATEDIFF(CurDate(), Employee_DateHired) AS 'Days With Company',
Employee.Employee_Name AS 'First Name', Employee.Employee_LastName AS
'Last Name'
```

```
FROM Employee
ORDER BY 'Days With Company' DESC;


END$$
DELIMITER ;

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `VAcustomers`()
BEGIN
/*
List of all customers that live in Virginia.
This list will be useful to employees because promotional emails and
advertisements can be sent to these customers since they’re more
likely to return than out of state visitors.



*/
SELECT Customer_Name As 'First Name', Customer_LastName AS 'Last
Name', Customer_Email AS 'Email' FROM Address, Customer WHERE
Customer.Customer_ID = Address.Customer_ID AND Address.State = 'VA';


END$$
DELIMITER ;

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `find_Sale`(IN `var_x`
VARCHAR(3))
BEGIN
/*
This query allows employees to see how many tickets were sold for a
particular movie showing.

*/
SELECT COUNT(Ticket.Movie_MovieID) AS `Tickets Sold`
FROM Ticket, Showing
WHERE Showing.Showing_ShowingID = Ticket.Showing_ShowingID AND
Ticket.Movie_MovieID = Showing.Movie_MovieID AND
Showing.Showing_ShowingID = var_x;

END$$
DELIMITER ;

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `numTickets`()
BEGIN
/*
Sum amount of tickets sold.
```

```
This query is important because it’d allow employees to see the amount
of tickets being sold.

*/
SELECT Sales.Sales_ItemType AS 'Type of Item',
SUM(Sales.Sales_QuantitySold) AS Sale
FROM Sales
WHERE Sales.Sales_ItemType = 'Ticket';


END$$
DELIMITER ;

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `receipt`()
BEGIN
/*
Create a receipt for customers.
This will be useful if a customer needs a receipt printed or sent to
them.

*/

SELECT Sales.Sales_SalesID AS 'Sales ID', Sales.Customer_ID AS
'Customer ID', Sales.Sales_Date, Sales_ItemType AS 'Type of Item',
CONCAT('$', FORMAT((Sales.Sales_Price), 2)) AS 'Sales Price',
Customer.Customer_Name AS 'Customer First Name',
Customer.Customer_LastName As 'Customer Last Name' FROM Sales INNER
JOIN Customer ON Sales.Customer_ID = Customer.Customer_ID;



END$$
DELIMITER ;

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `seats_Left`(IN `var_x`
VARCHAR(3))
BEGIN
/*
This query allows employees to see how many seats are remaining in a
theater for a specific showing of a movie.

*/
SELECT 50 - COUNT(Ticket.Movie_MovieID) AS `SeatsLeft`
FROM Ticket
WHERE Ticket.Showing_ShowingID = var_x;

END$$
DELIMITER ;
```

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `tickets_Sold`()
BEGIN
/*
Total dollar amount of sales for tickets.
This query is important because it’d show how much total money was
made on ticket sales.
*/
SELECT Sales.Sales_ItemType As 'Type of Item', CONCAT('$',
FORMAT(SUM(Sales.Sales_Price), 2)) AS Revenue
FROM Sales
WHERE Sales.Sales_ItemType = 'Ticket';


END$$
DELIMITER ;




/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```