BowlingAlley @TerpZone

Project_0504_10

Date – 03/11/2024

MEET OUR TEAM!

Piyali Bedagkar

Isha Tyagi

Gnapika Komaragiri

Samhitha Gontu

BACKGROUND

USERS

1. Students
2. Faculty
3. General Public

BOWLING EQUIPMENTS

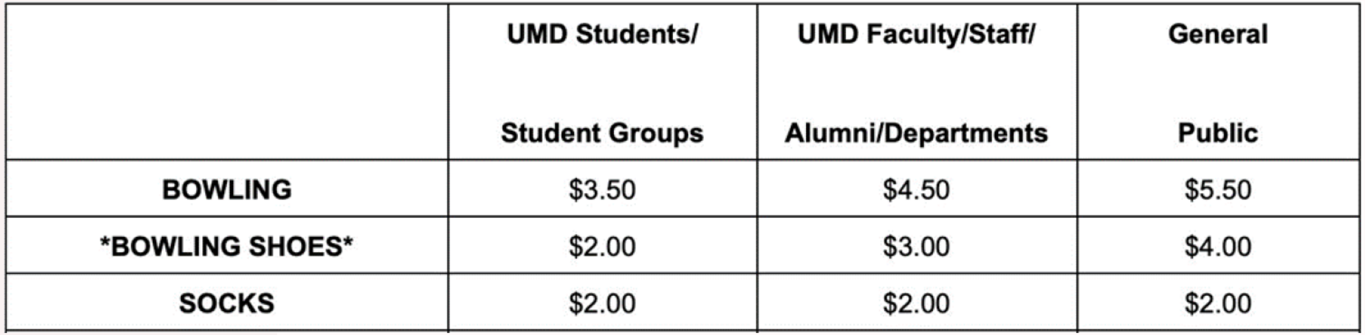
1. Bowling Alley
2. Bowling Shoes
3. Bowling Socks

DATA SOURCES

Our database project for TerpZone, the university bowling alley, encompasses data collected from two primary sources:

1. Customer surveys - The customer survey data was gathered through a week-long survey conducted with patrons visiting TerpZone. This dataset includes insights into customer demographics, preferences and frequency of visits.
2. Employee information - Additionally, employee information was obtained directly from the employees themselves and encompasses details such as their names, and any other pertinent data necessary for operational and managerial purposes.

These sources collectively provide a comprehensive foundation for our database, enabling us to analyze and optimize various aspects of TerpZone's operations and customer experiences.



	UMD Students/	UMD Faculty/Staff/	General
	Student Groups	Alumni/Departments	Public
BOWLING	$3.50	$4.50	$5.50
BOWLING SHOES	$2.00	$3.00	$4.00
SOCKS	$2.00	$2.00	$2.00

INTRODUCTION

Mission Statement

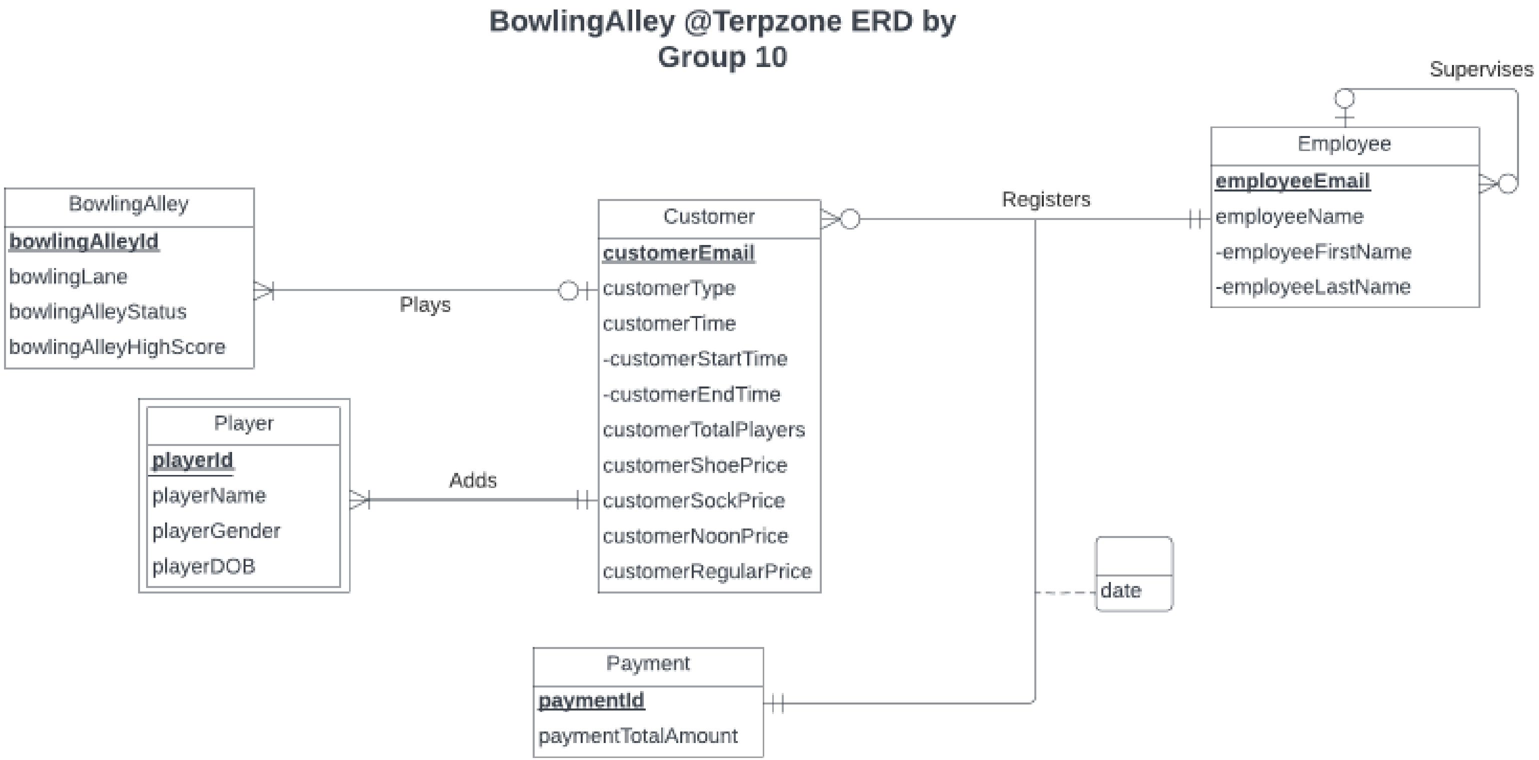
"To provide valuable insights and optimize the overall bowling experience for both customers and employees at Terpzone, we aim to establish a comprehensive database for the bowling alley. Our goal is to enhance operational efficiency seamlessly by integrating customer profiles, employee records, pricing structures, and transactional data at Terpzone."

INTRODUCTION

Mission Objectives

1. To analyze peak rush hours.
2. To leverage database insights to differentiate student, staff and general public demographics
3. To compare player engagement metrics between weekends and weekdays
4. To determine the predominant age group of players frequenting the visiting TerpZone BowlingAlley.

CONCEPTUAL DATABASE DESIGN: ER DIAGRAM

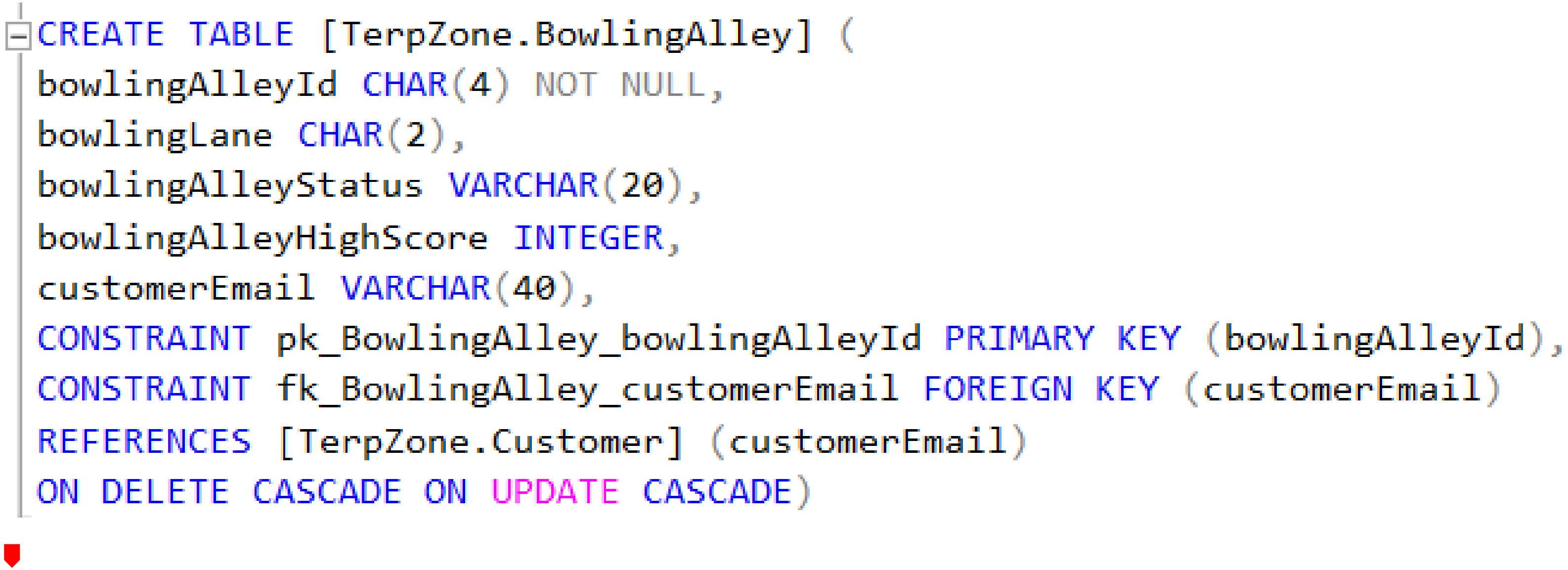


LOGICAL DATABASE DESIGN: RELATIONAL SCHEMA

- Employee(**employeeEmail**,employeeFirstname,employeeLastName,*employeeEmailSupervisor*)
- Customer(**customerEmail**, customerType, customerStartTime, customerEndTime, customerTotalPlayers, customerShoePrice, customerSockPrice, customerNoonPrice, customerRegularPrice)
- Payment(**paymentId**, paymentTotalAmount)
- Player(playerId, playerName, playerGender, playerDOB, *customerEmail*)
- BowlingAlley(**bowlingAlleyId**, bowlingLane, bowlingAlleyStatus, bowlingAlleyHighScore, *customerEmail*)
- Registers(***customerEmail***, *employeeEmail*, *paymentId*, date)

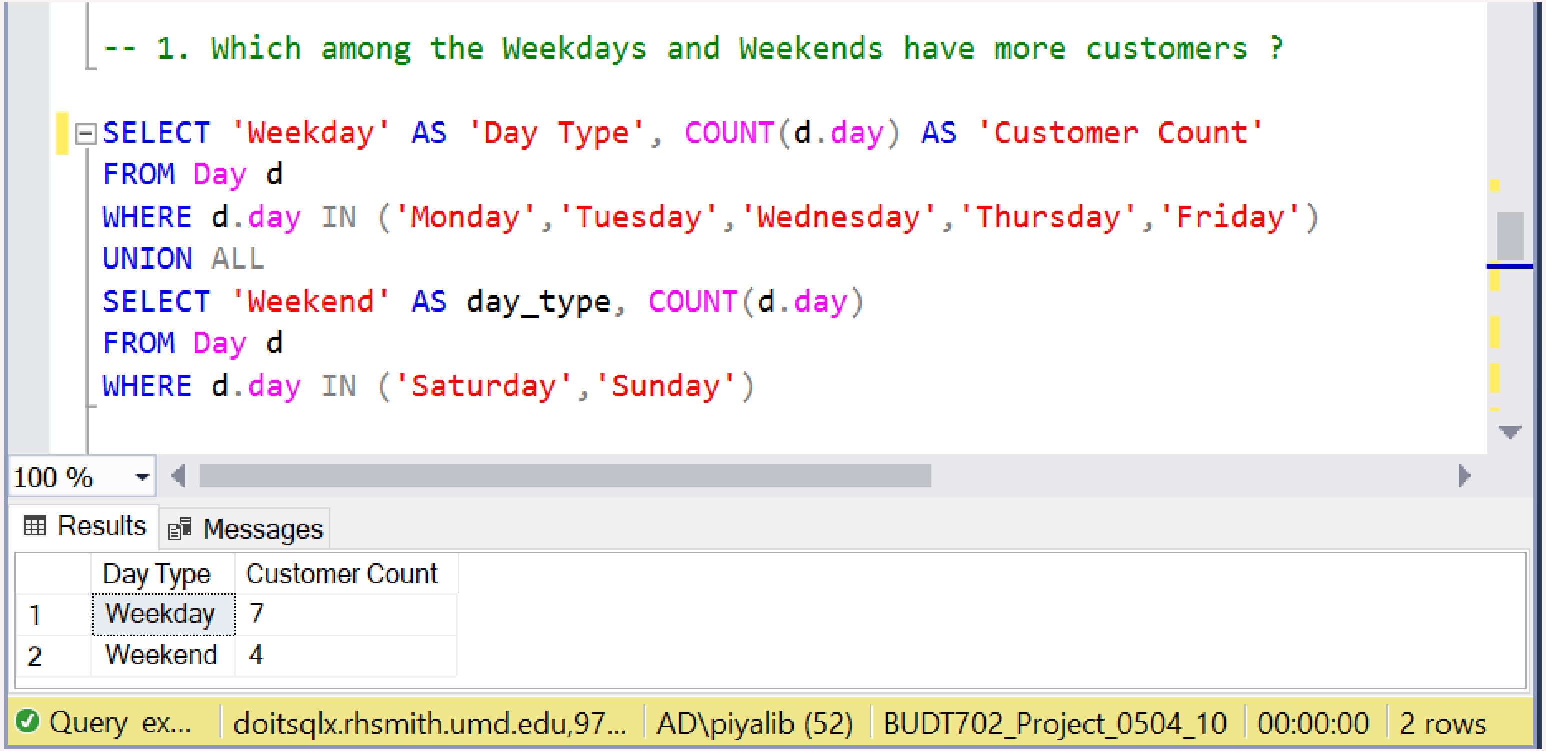
PHYSICAL DATABASE DESIGN

one SQL CREATE TABLE containing foreign key(s)



```sql
CREATE TABLE [TerpZone.BowlingAlley] (
bowlingAlleyId CHAR(4) NOT NULL,
bowlingLane CHAR(2),
bowlingAlleyStatus VARCHAR(20),
bowlingAlleyHighScore INTEGER,
customerEmail VARCHAR(40),
CONSTRAINT pk_BowlingAlley_bowlingAlleyId PRIMARY KEY (bowlingAlleyId),
CONSTRAINT fk_BowlingAlley_customerEmail FOREIGN KEY (customerEmail)
REFERENCES [TerpZone.Customer] (customerEmail)
ON DELETE CASCADE ON UPDATE CASCADE)
```

BUSINESS TRANSACTIONS



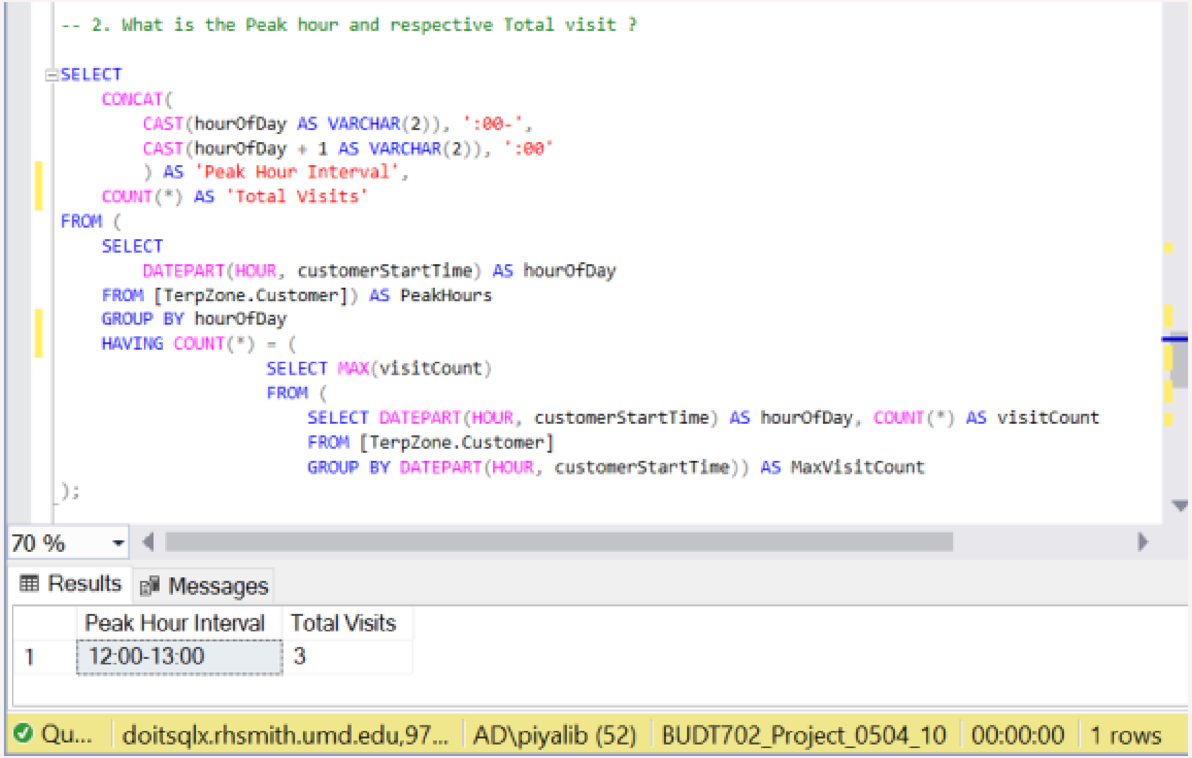
```
-- 1. Which among the Weekdays and Weekends have more customers ?

SELECT 'Weekday' AS 'Day Type', COUNT(d.day) AS 'Customer Count'
FROM Day d
WHERE d.day IN ('Monday','Tuesday','Wednesday','Thursday','Friday')
UNION ALL
SELECT 'Weekend' AS day_type, COUNT(d.day)
FROM Day d
WHERE d.day IN ('Saturday','Sunday')
```

	Day Type	Customer Count
1	Weekday	7
2	Weekend	4

Query ex... | doitsqlx.rhsmith.umd.edu,97... | AD\piyalib (52) | BUDT702_Project_0504_10 | 00:00:00 | 2 rows

BUSINESS TRANSACTIONS



```sql
-- 2. What is the Peak hour and respective Total visit ?

SELECT
    CONCAT(
        CAST(hourOfDay AS VARCHAR(2)), ':00-',
        CAST(hourOfDay + 1 AS VARCHAR(2)), ':00'
        ) AS 'Peak Hour Interval',
    COUNT(*) AS 'Total Visits'
FROM (
    SELECT
        DATEPART(HOUR, customerStartTime) AS hourOfDay
    FROM [TerpZone.Customer]) AS PeakHours
    GROUP BY hourOfDay
    HAVING COUNT(*) = (
                    SELECT MAX(visitCount)
                    FROM (
                        SELECT DATEPART(HOUR, customerStartTime) AS hourOfDay, COUNT(*) AS visitCount
                        FROM [TerpZone.Customer]
                        GROUP BY DATEPART(HOUR, customerStartTime)) AS MaxVisitCount
);
```

	Peak Hour Interval	Total Visits
1	12:00-13:00	3

```
Qu... | doitsqlx.rhsmith.umd.edu,97... | AD\piyalib (52) | BUDT702_Project_0504_10 | 00:00:00 | 1 rows
```

CONCLUSION

ENRICH THE BOWLING EXPERIENCE WITH TAILORED OPTIMIZATIONS FOR MAXIMUM FUN