# Final Project

Amusement Park Database

Aidan Carr
Marist College
November 21, 2022
CMPT308, Section 111

# Table of Contents

# Model Description

This database simulates day to day operations at the amusement park *Light Speed Park*. In this amusement park, there are two types of people: employees and guests. Each guest has their own unique ID number; the database stores this number, the guests' first and last name, their age, and height in inches. The database also stores information about employees like their employee ID number, first name, last name, age, and their employee position. Employees must be 14 or older to work here. Position is an entity used for identifying employee statuses. Each position has a unique position ID, position title, and a wage. This wage must be higher than or equal to the New Jersey minimum wage of $13. An employee can only hold one position at a time; these positions can be filled by zero to many employees.

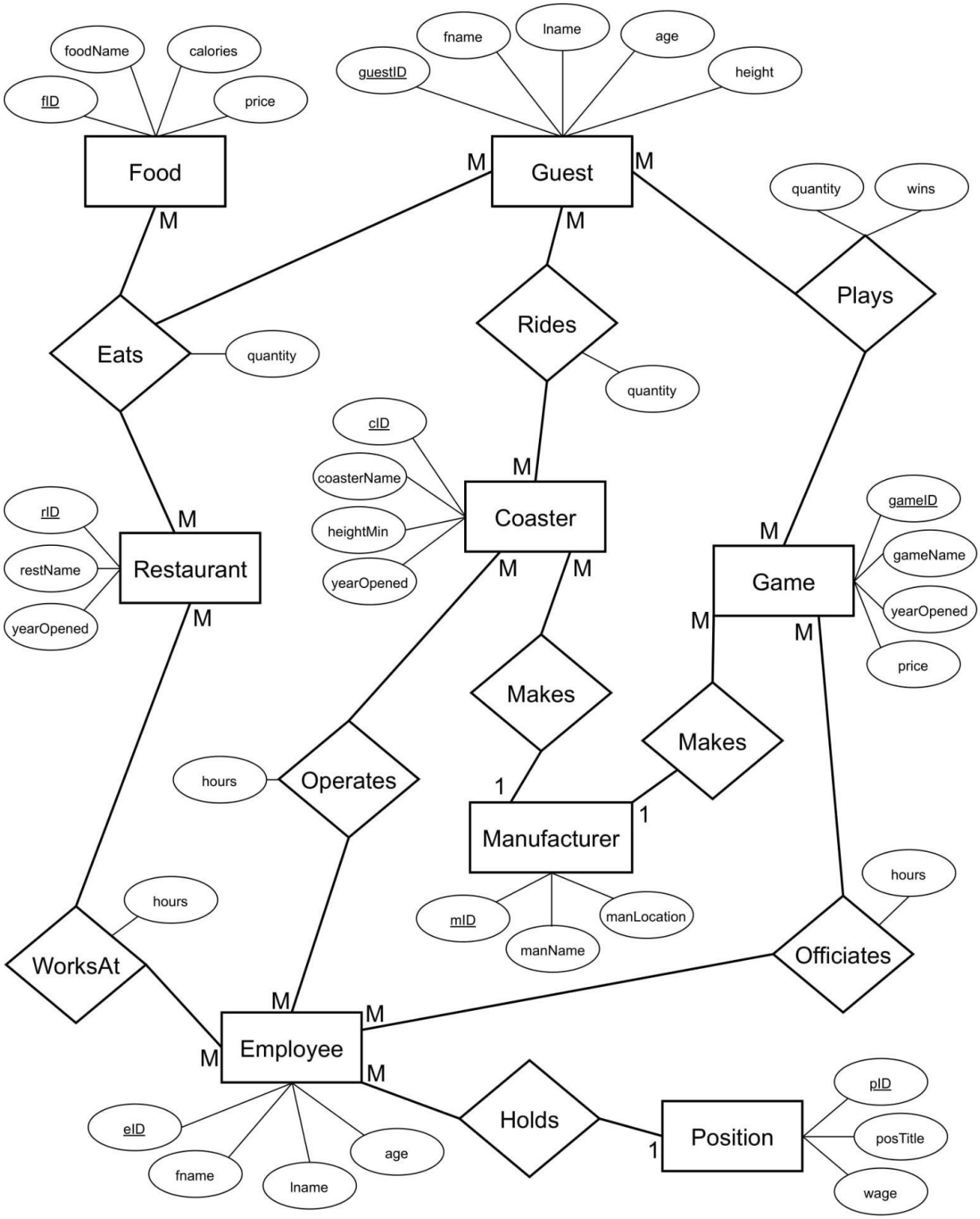
In this Amusement park, there are attractions grouped into three categories: Coasters, Restaurants, and Games. Restaurants have a unique ID, a name, and a year opened. Everytime a new restaurant is created, a new record is added. Any restaurants closed will remain in the database; rebranded restaurants will not overwrite the previous restaurant, but create a new record to keep previous guest transactions stored with that eatery. Coasters all have their unique ID, a name, a height requirement to ride in inches, and their year opened. Similar to restaurants, old coasters remain in the system and rebranded coasters become new records. Each coaster also has one manufacturer. A manufacturer has a unique ID, a name, and the location their headquarters resides. Games are one more attraction in the park; each one has a unique ID, a name, a year opened, and a price to play the game one time. Each game is made by one manufacturer. These manufacturers can produce many games and many coasters.

The database also stores the actions every employee and guest does in the park. The guests can eat food at a restaurant. Food is stored in the database with a unique ID, a name,

calorie count, and a price. This is a set price, so no matter where in the park the food is sold, it costs the guest the same amount of money. When eating, the guest's ID, food ID, and restaurant ID are stored. If a guest purchases multiple pieces of the same food, the stored quantity variable will increase. This variable stores the total amount of times this food item has been purchased by this customer at this particular restaurant. A guest can eat at many restaurants and eat many types of food. The restaurant can serve many types of food to many guests. Also, the food can be eaten by many guests and can be eaten at many restaurants. A guest can ride many coasters in the park, and each coaster can be ridden by many guests. After riding, the guest's ID and the coaster's ID are stored in the database, along with the total number of times this guest has ridden this ride. Finally, a guest can play a game, pairing both the guest ID and the game ID. Additionally, the quantity of times the person plays this game is stored and the number of wins is stored. These games can be played by many guests, and guests can play many different games.

Employees can work at many restaurants during their employment at the park, and each restaurant needs many employees working to keep it running. Every customer who works at a restaurant is stored in the database, along with the number of hours they work there. Similarly, employees operate one to many coasters; coasters are operated by many workers. When an employee operates a coaster, their ID, the coaster's ID, and the number of total hours are stored. Employees also officiate games; they can officiate many games and games can be officiated by many employees. Everytime a worker officiates a game, the employee ID, game ID, and hours are stored. When an employee works on the same game again, the hours add on top of the total hours— this rule similarly applies to employees working at a restaurant or a coaster.

# ER Diagram


foodName
calories
fID
price
Food
fname
lname
age
guestID
height
Guest
quantity
wins
Plays
Eats
quantity
Rides
quantity
cID
coasterName
heightMin
yearOpened
Coaster
rID
restName
yearOpened
Restaurant
gameID
gameName
yearOpened
price
Game
Makes
Makes
hours
Operates
Manufacturer
mID
manName
manLocation
hours
WorksAt
hours
Officiates
Employee
eID
fname
lname
age
Holds
Position
pID
posTitle
wage
M
1

# Guest Table

```
CREATE TABLE oGUEST
(
  GUESTID NUMBER NOT NULL
, FNAME VARCHAR2(20) NOT NULL
, LNAME VARCHAR2(30) NOT NULL
, AGE NUMBER NOT NULL
, HEIGHT NUMBER NOT NULL
);

ALTER TABLE oGUEST
ADD CONSTRAINT PK_GUEST PRIMARY KEY (GUESTID);
```

This table represents guests at Light Speed Park; each record of the table represents one guest. A guest is identified by the primary key: `guestID`. Other attributes of the table include their first name: `fname`, last name: `lname`, age: `age`, and height in inches: `height`.

# Employee Table

```
CREATE TABLE oEMPLOYEE
(
  EID NUMBER NOT NULL
, FNAME VARCHAR2(20) NOT NULL
, LNAME VARCHAR2(30) NOT NULL
, AGE NUMBER NOT NULL
, PID NUMBER NOT NULL
);

ALTER TABLE oEMPLOYEE
ADD CONSTRAINT PK_EMPLOYEE PRIMARY KEY (EID);

ALTER TABLE oEMPLOYEE
ADD CONSTRAINT FK_EMPLOYEE_POSITION FOREIGN KEY (PID) REFERENCES
oPOSITION
ADD CONSTRAINT EMPLOYEE_CK1 CHECK(AGE >=14);
```

This table represents employees at Light Speed Park; each record of the table represents one employee. An employee is identified by the primary key: `eID`. Other attributes of the table include their first name: `fname`, last name: `lname`, age: `age`, and position ID: `pID`. The attribute `pID` is a foreign key that references the primary key of the Position table, the one position the employee holds. The attribute `age` has a constraint where it has to be greater than or equal to 14, meaning the employee can only exist if they are 14 or older.

# Position Table

```
CREATE TABLE oPOSITION
(
  PID NUMBER NOT NULL
, POSTITLE VARCHAR2(30) NOT NULL
, WAGE NUMBER NOT NULL
);

ALTER TABLE oPOSITION
ADD CONSTRAINT PK_POSITION PRIMARY KEY (PID)
ADD CONSTRAINT POSITION_CK1 CHECK(WAGE >= 13);
```

This table represents employees' positions at Light Speed Park; each record of the table represents one position. A position is identified by the primary key `pID`. Other attributes of the table include position title/name: `posTitle`, and wage: `wage`. The attribute `wage` has a constraint where it has to be greater than or equal to 13, meaning every position must pay greater than or equal to $13 an hour.

# Restaurant Table

```
CREATE TABLE oRESTAURANT
(
  RID NUMBER NOT NULL
, RESTNAME VARCHAR2(30) NOT NULL
, YEAROPENED NUMBER NOT NULL
);

ALTER TABLE oRESTAURANT
ADD CONSTRAINT PK_RESTAURANT PRIMARY KEY (RID);
```

This table represents restaurants at Light Speed Park; each record of the table represents one restaurant. A restaurant is identified by the primary key: `rID`. Other attributes of the table include restaurant name: `restName`, and yearOpened: `yearOpened`.

# Coaster Table

```
CREATE TABLE oCOASTER
(
  CID NUMBER NOT NULL
, COASTERNAME VARCHAR2(30) NOT NULL
, HEIGHTMIN NUMBER NOT NULL
, YEAROPENED NUMBER NOT NULL
, MID NUMBER NOT NULL
);

ALTER TABLE oCOASTER
ADD CONSTRAINT PK_COASTER PRIMARY KEY (CID);

ALTER TABLE oCOASTER
ADD CONSTRAINT FK_COASTER_MANUFACTURER FOREIGN KEY (MID)
REFERENCES oMANUFACTURER;
```

This table represents coasters at Light Speed Park; each record of the table represents one coaster. A coaster is identified by the primary key: `cID`. Other attributes of the table include the coaster name: `coasterName`, height minimum requirement to ride in inches: `heightMin`, year opened: `yearOpened`, and manufacturer ID: `mID`. The attribute `mID` is a foreign key that references the primary key of the Manufacturer table, the one manufacturer that manufactures a coaster.

# Game Table

```
CREATE TABLE oGAME
(
  GAMEID NUMBER NOT NULL
, GAMENAME VARCHAR2(30) NOT NULL
, YEAROPENED NUMBER NOT NULL
, PRICE NUMBER NOT NULL
, MID NUMBER NOT NULL
);

ALTER TABLE oGAME
ADD CONSTRAINT PK_GAME PRIMARY KEY (GAMEID);

ALTER TABLE oGAME
ADD CONSTRAINT FK_GAME_MANUFACTURER FOREIGN KEY (MID) REFERENCES
oMANUFACTURER;
```

This table represents games at Light Speed Park; each record of the table represents one game. A game is identified by the primary key: `gameID`. Other attributes of the table include the game name: `gameName`, year opened: `yearOpened`, price to play: `price`, and manufacturer ID: `mID`. The attribute `mID` is a foreign key that references the primary key of the Manufacturer table, the one manufacturer that manufactures a game.

# Food Table

```
CREATE TABLE oFOOD
(
  FID NUMBER NOT NULL
, FOODNAME VARCHAR2(30) NOT NULL
, CALORIES NUMBER NOT NULL
, PRICE NUMBER NOT NULL
);

ALTER TABLE oFOOD
ADD CONSTRAINT PK_FOOD PRIMARY KEY (FID);
```

This table represents food at Light Speed Park; each record of the table represents one food item. A food is identified by the primary key: `fID`. Other attributes of the table include the food name: `foodName`, calorie count: `calories`, and price to purchase: `price`.

# Manufacturer Table

```
CREATE TABLE oMANUFACTURER
(
  MID NUMBER NOT NULL
, MANNAME VARCHAR2(30) NOT NULL
, MANLOCATION VARCHAR2(30) NOT NULL
);

ALTER TABLE oMANUFACTURER
ADD CONSTRAINT PK_MANUFACTURER PRIMARY KEY (MID);
```

This table represents manufacturers for attractions at Light Speed Park; each record of the table represents one manufacturer. A manufacturer is identified by the primary key: `mID`. Other attributes of the table include the manufacturer name: `manName`, and the manufacturer's location: `manLocation`.

# Plays Table

```
CREATE TABLE oPLAYS
(
  GUESTID NUMBER NOT NULL
, GAMEID NUMBER NOT NULL
, QUANTITY NUMBER NOT NULL
, WINS NUMBER NOT NULL
);

ALTER TABLE oPLAYS
ADD CONSTRAINT PK_PLAYS PRIMARY KEY (GUESTID, GAMEID);

ALTER TABLE oPLAYS
ADD CONSTRAINT FK_PLAYS_GUEST FOREIGN KEY (GUESTID) REFERENCES
oGUEST
ADD CONSTRAINT FK_PLAYS_GAME FOREIGN KEY (GAMEID) REFERENCES
oGAME;
```

This table represents games played by guests at Light Speed Park; each record of the table represents a certain guest who plays a certain game. An instance of play is identified by the primary key: `guestID` and `gameID`. Other attributes of the table include the number of times played at this location for this person: `quantity`, and the number of times won at this location for this person: `wins`. The attribute `guestID` is a foreign key that references the primary key of the Guest table. The attribute `gameID` is a foreign key that references the primary key of the Game table.

# Rides Table

```
CREATE TABLE oRIDES
(
  GUESTID NUMBER NOT NULL
, CID NUMBER NOT NULL
, QUANTITY NUMBER NOT NULL
);

ALTER TABLE oRIDES
ADD CONSTRAINT PK_RIDES PRIMARY KEY (GUESTID, CID);

ALTER TABLE oRIDES
ADD CONSTRAINT FK_RIDES_GUEST FOREIGN KEY (GUESTID) REFERENCES
oGUEST
ADD CONSTRAINT FK_RIDES_GAME FOREIGN KEY (CID) REFERENCES
oCOASTER;
```

This table represents coasters ridden by guests at Light Speed Park; each record of the table represents a certain guest who rides a certain coaster. An instance of ride is identified by the primary key: `guestID` and `cID`. Other attributes of the table include the total times ridden at this location for this person: `quantity`. The attribute `guestID` is a foreign key that references the primary key of the Guest table. The attribute `cID` is a foreign key that references the primary key of the Coaster table.

# Eats Table

```
CREATE TABLE oEATS
(
  GUESTID NUMBER NOT NULL
, FID NUMBER NOT NULL
, RID NUMBER NOT NULL
, QUANTITY NUMBER NOT NULL
);

ALTER TABLE oEATS
ADD CONSTRAINT PK_EATS PRIMARY KEY (GUESTID, FID, RID);

ALTER TABLE oEATS
ADD CONSTRAINT FK_EATS_GUEST FOREIGN KEY (GUESTID) REFERENCES
oGUEST
ADD CONSTRAINT FK_EATS_FOOD FOREIGN KEY (FID) REFERENCES oFOOD
ADD CONSTRAINT FK_EATS_RESTAURANT FOREIGN KEY (RID) REFERENCES
oRESTAURANT;
```

This table represents foods eaten by guests in restaurants at Light Speed Park; each record of the table represents a certain guest who eats a certain food at a certain restaurant. An instance of eat is identified by the primary key: `guestID`, `fID`, and `rID`. Other attributes of the table include the total amount this food ordered at this location for this person: `quantity`. The attribute `guestID` is a foreign key that references the primary key of the Guest table. The attribute `fID` is a foreign key that references the primary key of the Food table. The attribute `rID` is a foreign key that references the primary key of the Restaurant table.

# Works At Table

```
CREATE TABLE oWORKSAT
(
  EID NUMBER NOT NULL
, RID NUMBER NOT NULL
, HOURS NUMBER NOT NULL
);
ALTER TABLE oWORKSAT
ADD CONSTRAINT PK_WORKSAT PRIMARY KEY (EID, RID);

ALTER TABLE oWORKSAT
ADD CONSTRAINT FK_WORKSAT_EMPLOYEE FOREIGN KEY (EID) REFERENCES
oEMPLOYEE
ADD CONSTRAINT FK_WORKSAT_RESTAURANT FOREIGN KEY (RID)
REFERENCES oRESTAURANT;
```

This table represents employees working at restaurants at Light Speed Park; each record of the table represents a certain employee who works at a certain restaurant. An instance of working is identified by the primary key: `eID` and `rID`. Other attributes of the table include the number of total hours worked at this location for this person: `hours`. The attribute `eID` is a foreign key that references the primary key of the Employee table. The attribute `rID` is a foreign key that references the primary key of the Restaurant table.

# Operates Table

```
CREATE TABLE oOPERATES
(
  EID NUMBER NOT NULL
, CID NUMBER NOT NULL
, HOURS NUMBER NOT NULL
);

ALTER TABLE oOPERATES
ADD CONSTRAINT PK_OPERATES PRIMARY KEY (EID, CID);

ALTER TABLE oOPERATES
ADD CONSTRAINT FK_OPERATES_EMPLOYEE FOREIGN KEY (EID) REFERENCES
oEMPLOYEE
ADD CONSTRAINT FK_OPERATES_COASTER FOREIGN KEY (CID) REFERENCES
oCOASTER;
```

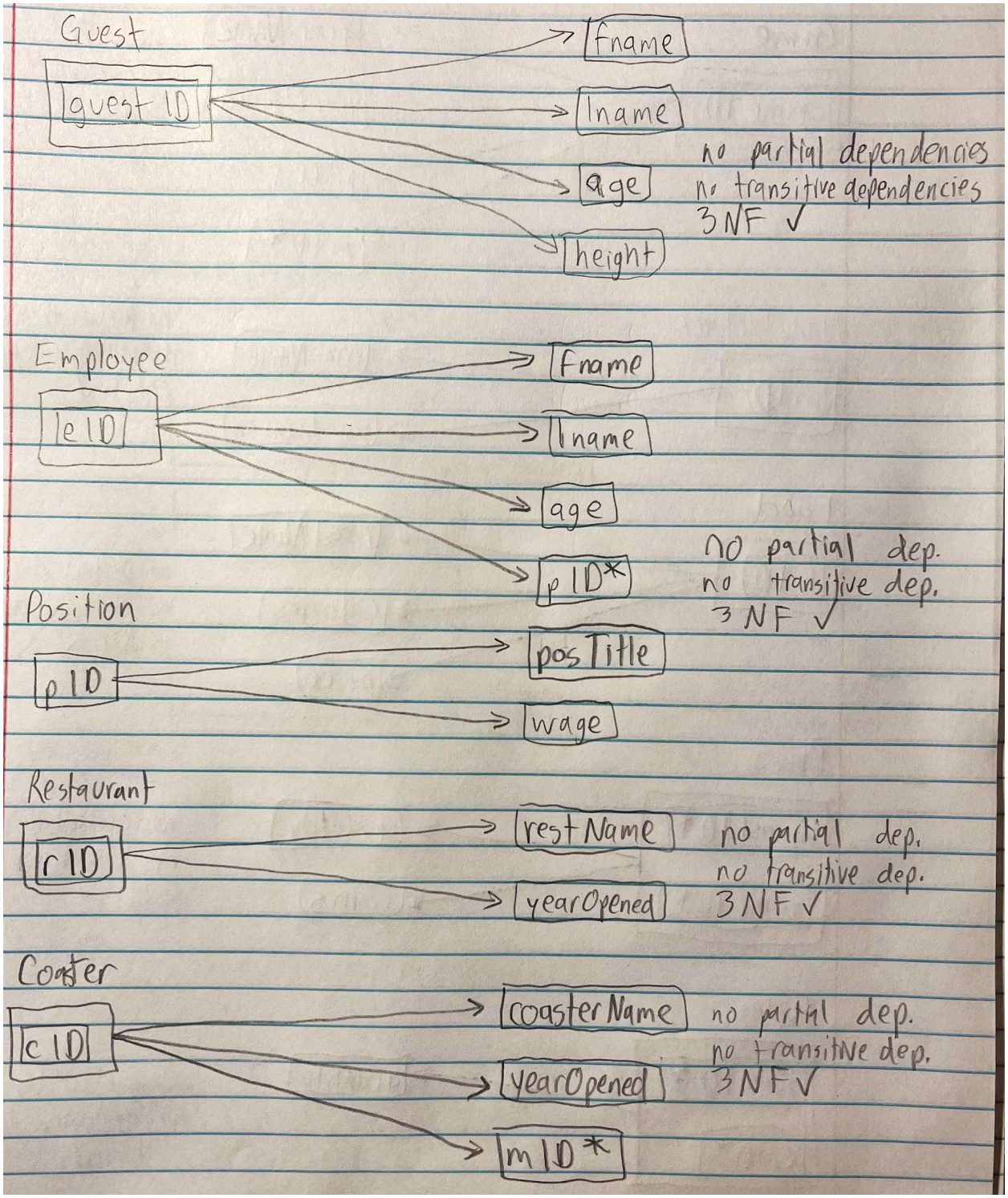
This table represents employees operating coasters at Light Speed Park; each record of the table represents a certain employee who operates a certain coaster. An instance of operating is identified by the primary key: `eID` and `cID`. Other attributes of the table include the number of total hours worked at this location for this person: `hours`. The attribute `eID` is a foreign key that references the primary key of the Employee table. The attribute `cID` is a foreign key that references the primary key of the Coaster table.

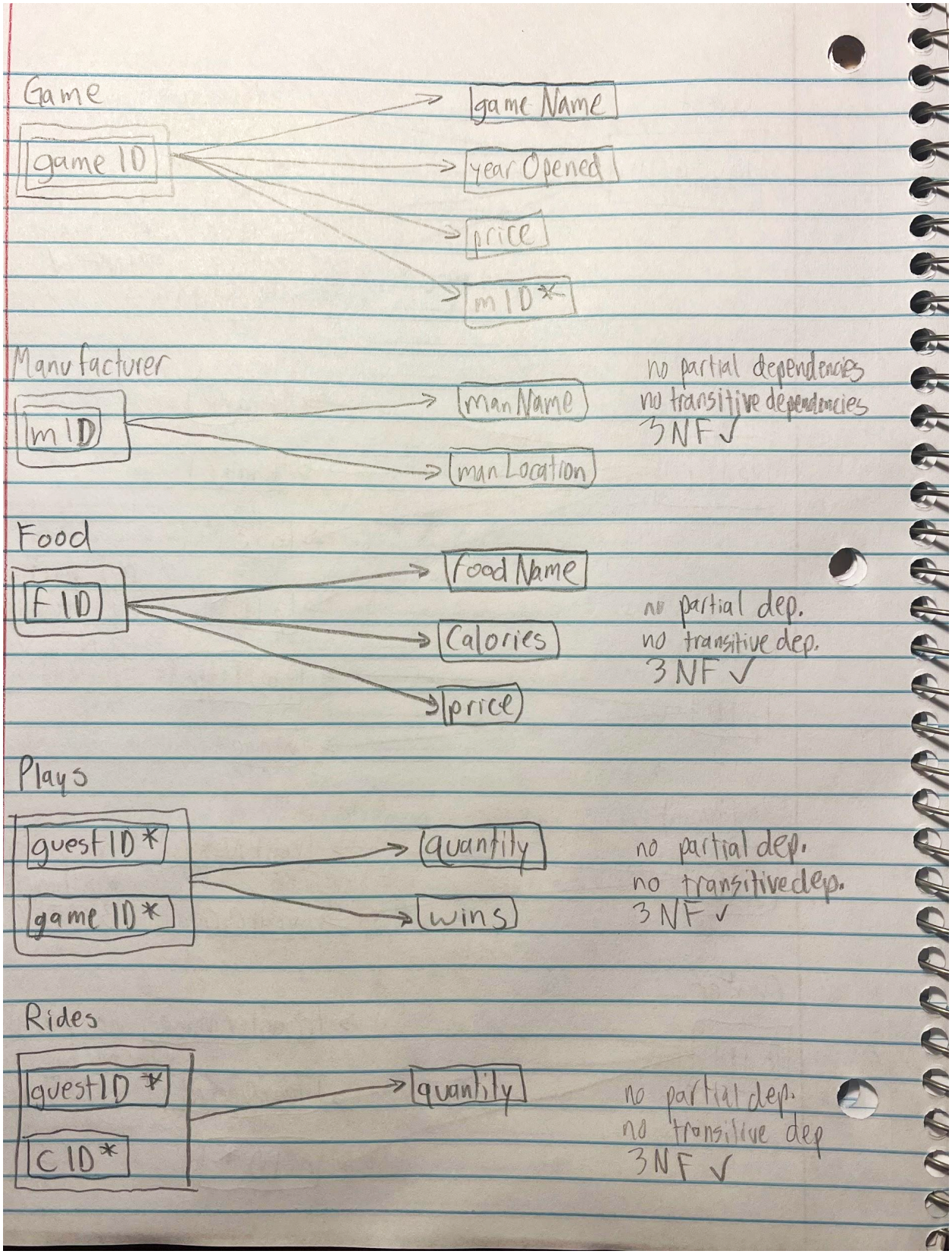
# Officiates Table

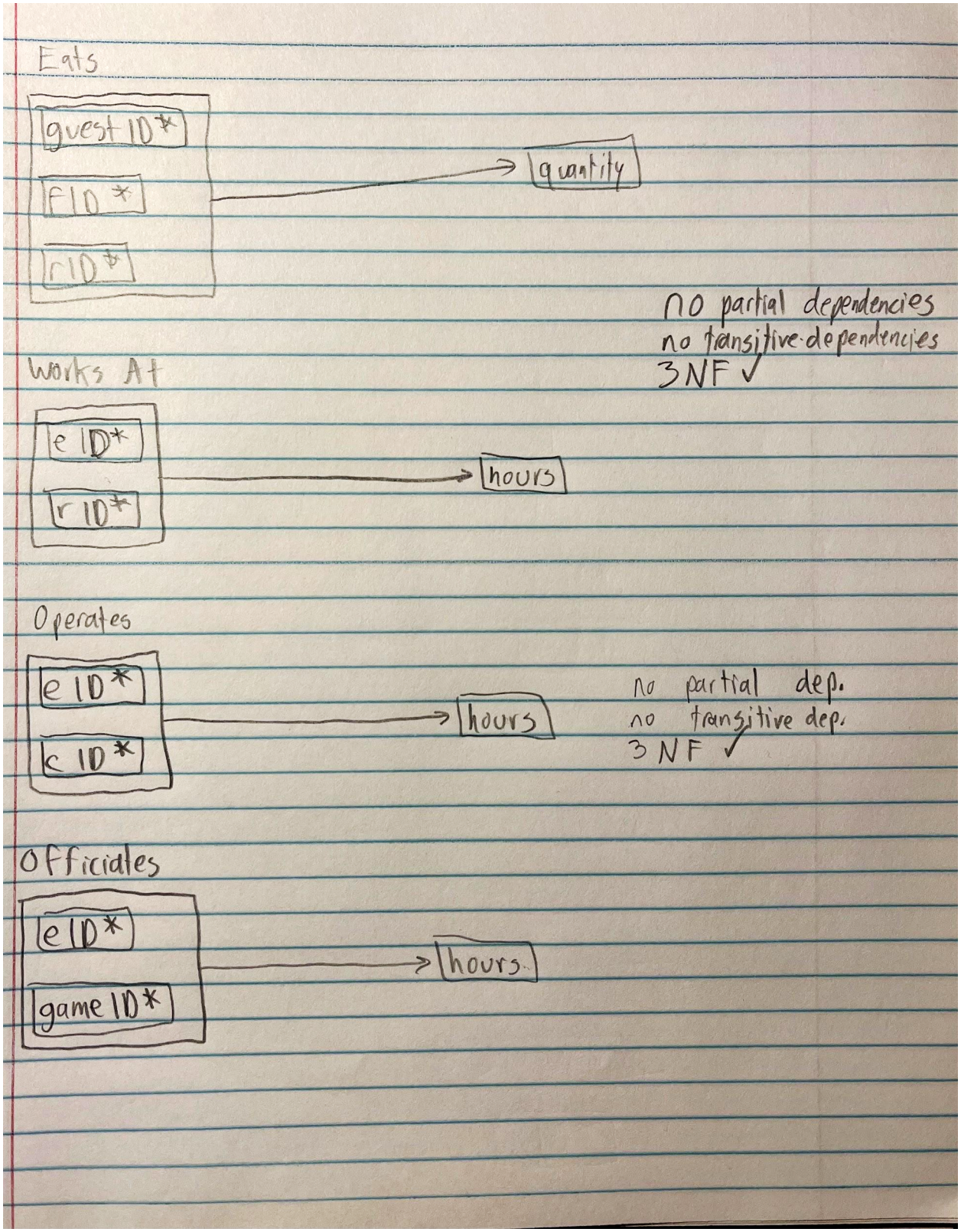
```
CREATE TABLE oOFFICIATES
(
  EID NUMBER NOT NULL
, GAMEID NUMBER NOT NULL
, HOURS NUMBER NOT NULL
);

ALTER TABLE oOFFICIATES
ADD CONSTRAINT PK_OFFICIATES PRIMARY KEY (EID, GAMEID);

ALTER TABLE oOFFICIATES
ADD CONSTRAINT FK_OFFICIATES_EMPLOYEE FOREIGN KEY (EID)
REFERENCES oEMPLOYEE
ADD CONSTRAINT FK_OFFICIATES_GAME FOREIGN KEY (GAMEID)
REFERENCES oGAME;
```

This table represents employees officiating games at Light Speed Park; each record of the table represents a certain employee who officiates a certain game. An instance of officiating is identified by the primary key: `eID` and `gameID`. Other attributes of the table include the number of total hours worked at this location for this person: `hours`. The attribute `eID` is a foreign key that references the primary key of the Employee table. The attribute `gameID` is a foreign key that references the primary key of the Game table.

# Justification of Normal Form




Game
game ID
game Name
year Opened
price
mID*
Manufacturer
mID
manName
manLocation
no partial dependencies
no transitive dependencies
3NF ✓
Food
FID
Food Name
Calories
price
no partial dep.
no transitive dep.
3NF ✓
Plays
guestID*
game ID*
quantity
wins
no partial dep.
no transitive dep.
3NF ✓
Rides
guestID*
CID*
quantity
no partial dep.
no transitive dep
3NF ✓


Eats
guest ID*
FID*
rID*
quantity
no partial dependencies
no transitive dependencies
3NF ✓
Works At
eID*
rID*
hours
Operates
eID*
cID*
hours
no partial dep.
no transitive dep.
3NF ✓
Officiates
eID*
gameID*
hours

**oGuest**
guestID, fname, lname, age, height
PK: {guestID}
FK: Ø

**oEmployee**
eID, fname, lname, age, pID
PK: {eID}
FK: {pID} references oPosition

**oPosition**
pID, posTitle, wage
PK: {pID}
FK: Ø

**oRestaurant**
rID, restName, yearOpened
PK: {rID}
FK: Ø

**oCoaster**
cID, coasterName, heightMin yearOpened, mID
PK: {cID}
FK: {mID} references oManufacturer

**oGame**
gameID, gameName, yearOpened, price, mID
PK: {gameID}
FK: {mID} references oManufacturer

**oFood**
fID, foodName, calories, price
PK: {fID}
FK: Ø

**oManufacturer**
mID, manName, manLocation
PK: {mID}
FK: Ø

**oPlays**
guestID, gameID, quantity, wins
PK: {guestID, gameID}
FK: {guestID} references oGuest
FK: {gameID} references oGame

**oRides**
guestID, cID, quantity
PK: {guestID, cID}
FK: {guestID} references oGuest
FK: {cID} references oCoaster

**oEats**
guestID, fID, rID, quantity
PK: {guestID, fID, rID}
FK: {guestID} references oGuest
FK: {fID} references oFood
FK: {rID} references oRestaurant

**oWorksAt**
eID, rID, hours
PK: {eID, rID}
FK: {eID} references oEmployee
FK: {rID} references oRestaurant

**oOperates**
eID, cID, hours
PK: {eID, cID}
FK: {eID} references oEmployee
FK: {cID} references oCoaster

**oOfficiates**
eID, gameID, hours
PK: {eID, gameID}
FK: {eID} references oEmployee
FK: {gameID} references oGame

# Query 1

Give full names of people who have ridden every ride that opened in 2007

```
CREATE OR REPLACE VIEW FINQUERY1 AS
SELECT  G.fname, G.lname
FROM    oGuest G
WHERE   NOT EXISTS
    (SELECT *
    FROM    oCoaster C
    WHERE   C.yearOpened = 2007
    AND     NOT EXISTS
        (SELECT *
        FROM    oRides R
        WHERE   G.guestID = R.guestID
        AND     R.cid = C.cid) ) ;
```

```
FNAME                LNAME
-------------------- ------------------------------
Robert               Davis
Kasey                Walker
Kay                  McNeil

Cardinality of 3
```

# Query 2

Name restaurants who have only served people who are taller than 5 feet

```
CREATE OR REPLACE VIEW FINQUERY2 AS
SELECT  R.restName
FROM    oRestaurant R
WHERE   R.rID NOT IN
    (SELECT E.rID
    FROM    oEats E
    WHERE   E.guestID NOT IN
        (SELECT G.guestID
        FROM    oGuest G
        WHERE   G.height > 60) ) ;

RESTNAME
------------------------------
The Screaming Plate
Brave Bowl
Johns Grill
Aunt Ellies Diner

Cardinality of 4
```

# Query 3

List guest IDs and ages of the teenagers (13-17 year olds) who have not played any games

```
CREATE OR REPLACE VIEW FINQUERY3 AS
SELECT  G.guestID, G.age
FROM    oGuest G
WHERE   G.age BETWEEN 13 AND 17
AND     G.guestID NOT IN
    (SELECT P.guestID
    FROM    oPlays P
    WHERE   P.gameID IN
        (SELECT G.gameID
        FROM    oGame G ) ) ;
```

| GUESTID | AGE |
|---|---|
| 105 | 16 |
| 111 | 13 |
| 114 | 17 |
| 126 | 15 |
| 129 | 14 |
| 131 | 16 |

Cardinality of 6

# Query 4

Name titles of all positions and the employee IDs of those holding that position, if any

```
CREATE OR REPLACE VIEW FINQUERY4 AS
SELECT  P.posTitle, e.eID
FROM    oPosition P LEFT JOIN oEmployee E
ON      P.pID = E.pID;
```

```
POSTITLE                          EID
------------------------------ ----------
Games Supervisor                    5026
Games Supervisor                    5046
Games Supervisor                    5047
Restaurant Manager                  5035
Restaurant Manager                  5036
Restaurant Manager                  5028
Coaster Manager                     5041
Coaster Manager                     5020
Games Manager                       5040
Games Manager                       5013
Games Manager                       5004
Team Scheduler                      5015
Park Relations Manager              5005
Park Relations Manager              5045
Security Officer                    5016
Security Officer                    5050
Security Officer                    5011
Technologies Technician             5025
Technologies Technician             5001
Social Media Advisor

Cardinality of 51, final 20 records shown here
```

# Query 5

Name all manufacturers along with the coasters they have manufactured, if any

```
CREATE OR REPLACE VIEW FINQUERY5 AS
SELECT  C.coasterName, M.manName
FROM    oCoaster C RIGHT JOIN oManufacturer M
ON      C.mID = M.mID;
```

| COASTERNAME | MANNAME |
|---|---|
| Karate Simulation | Bob the Builder |
| Mount Doom | Bob the Builder |
| El Toro | Bob the Builder |
| | The Scam Artist |
| Mr Beans Drive | Rickety Metal |
| Snake Ride | Rickety Metal |
| Dumbos Flight | Wheels on Wheels |
| Lighting Loops | Wheels on Wheels |
| Techno Beam | Wheels on Wheels |
| Medusa | Wheels on Wheels |
| Parachutes | Wheels on Wheels |
| Swings of Death | Iron Man Construction |
| Runaway Train | Iron Man Construction |
| Soarin | Iron Man Construction |
| | Franklin Morris LLC |
| Race to Ohio | Toddlers With Tools |
| Bizzaro | Toddlers With Tools |
| Rocky Flight | Toddlers With Tools |
| Rolling Thunder | Toddlers With Tools |
| | Sports Brothers |
| Bashi Bashi | Alien Architectures |
| Jolly Rodger | Alien Architectures |
| Flight of the Banshee | Alien Architectures |
| Scream Machine | Alien Architectures |

Cardinality of 24

# Query 6

Name all games and the year they opened, along with the names of restaurants with the same opening year (if any), along with any remaining restaurants

```
CREATE OR REPLACE VIEW FINQUERY6 AS
SELECT  G.gameName, G.yearOpened, R.restName
FROM    oGame G FULL JOIN oRestaurant R
ON      G.yearOpened = R.yearOpened;
```

| GAMENAME | YEAROPENED | RESTNAME |
|---|---|---|
| | | Grillville |
| Pistol Race | 2007 | Toms Shack |
| Bounce and Toss | 2007 | Toms Shack |
| | | Bev Zone |
| Sloth Vacation | 2020 | Treat Place |
| | | Fried Goodness |
| | | The Brew House |
| Pin Knockdown | 2003 | Ryans Cafe |
| Slam or Slide | 2019 | Quick Eats |
| Laser Battle | 2014 | Fill Yur Belly |
| Pistol Race | 2007 | The Screaming Plate |
| Bounce and Toss | 2007 | The Screaming Plate |
| | | Brave Bowl |
| | | Johns Grill |
| Climberz | 1996 | Aunt Ellies Diner |
| Balls on Walls | 1991 | |
| Green Day Slots | 2017 | |
| Basketball Brawl | 1997 | |
| Ring to Win | 2002 | |

Cardinality of 19

# Query 7

List the first name and ages of guests who have either:

a. Eaten a salad
b. Ridden a coaster with a minimum height requirement of 48 inches
c. Won one game at least 2 times

```
CREATE OR REPLACE VIEW FINQUERY7 AS
SELECT  G1.fname, G1.age
FROM    oGuest G1, oEats E1, oFood F1
WHERE   G1.guestID = E1.guestID
AND     E1.fid = F1.fid
AND     F1.foodName = 'Salad'

UNION

SELECT  G2.fname, G2.age
FROM    oGuest G2, oRides R2, oCoaster C2
WHERE   G2.guestID = R2.guestID
AND     R2.cID = C2.cID
AND     C2.heightMin = 48

UNION

SELECT  G3.fname, G3.age
FROM    oGuest G3, oPlays P3
WHERE   G3.guestID = P3.guestID
AND     P3.wins >= 2 ;
```

| FNAME | AGE |
|---|---|
| Ana | 12 |
| Ben | 77 |
| Carl | 34 |
| Carly | 17 |
| Christian | 13 |
| Christina | 23 |
| Donald | 68 |
| Emily | 17 |
| Freddy | 25 |
| Heather | 54 |
| Jim | 34 |
| Lucy | 8 |
| Martha | 86 |
| Michael | 24 |
| Rachel | 56 |
| Robert | 53 |
| Ronald | 13 |
| Rose | 24 |
| Samuel | 19 |
| TJ | 16 |
| Tiffany | 8 |
| Todd | 57 |
| Tyler | 90 |

Cardinality of 23

# Query 8

For each employee that officiates over 10 hours, list their ID and the total hours they've worked across all games

```
CREATE OR REPLACE VIEW FINQUERY8 AS
SELECT   O.eID, SUM(O.hours) AS total_hours
FROM     oOfficiates O
GROUP BY (eID)
HAVING   SUM(O.hours) > 10;
```

| EID | TOTAL_HOURS |
|---|---|
| 5001 | 467.5 |
| 5004 | 268.9 |
| 5011 | 233.5 |
| 5013 | 363.1 |
| 5016 | 590.9 |
| 5022 | 112.1 |
| 5024 | 67.6 |
| 5025 | 234.3 |
| 5026 | 194.4 |
| 5033 | 28.8 |
| 5040 | 466.8 |
| 5042 | 108.7 |
| 5044 | 43 |
| 5046 | 194 |
| 5047 | 175 |
| 5050 | 253.4 |

Cardinality of 16

# Query 9

Give full names and position titles for employees who have operated a coaster that opened before the year 2000

```
CREATE OR REPLACE VIEW FINQUERY9 AS
SELECT  DISTINCT E.fname, E.lname, P.posTitle
FROM    oEmployee E, oOperates O, oCoaster C, oPosition P
WHERE   P.pID = E.pID
AND     E.eID = O.eID
AND     O.cID = C.cid
AND     C.yearOpened < 2000;
```

| FNAME | LNAME | POSTITLE |
|---|---|---|
| Mia | Flores | Coaster Team Member |
| Carolyn | Parker | Coaster Team Member |
| Sunny | Lee | Coaster Supervisor |
| Deb | Patel | Coaster Manager |
| DJ | Hall | Security Officer |
| Eric | Reed | Coaster Team Member |
| Kevin | White | Coaster Supervisor |
| Brian | Lee | Coaster Supervisor |

Cardinality of 8

# Query 10

Name restaurants where no minors (employees younger than 18) have worked

```
CREATE OR REPLACE VIEW FINQUERY10 AS
SELECT  R.restName
FROM    oRestaurant R
WHERE   NOT EXISTS
    (SELECT *
    FROM    oEmployee E
    WHERE   E.age < 18
    AND     EXISTS
        (SELECT *
        FROM    oWorksAt W
        WHERE   W.rID = R.rID
        AND     W.eID = E.eID ) ) ;

RESTNAME
------------------------------
Grillville
Toms Shack
Quick Eats
Johns Grill

Cardinality of 4
```