THE ULTIMATE SUMMER FESTIVAL APP

by Andreea Atanasescu & Kelly Luu

WHAT IS THE PURPOSE OF OUR APP?

- it operates based on a database that stores all the necessary information for both organising and attending a summer festival
- it has easy-to-use features that offer a complete festival experience to all its users
- it is fun, because we more than surely had fun creating it

WHO CAN USE THE APP?

THE PARTICIPANTS:

- who can check for updates and festival information

THE STAFF:

- who can access parts of the database to facilitate event organisation

NOW...

LET’S MOVE ON TO SOME LOGISTICS!

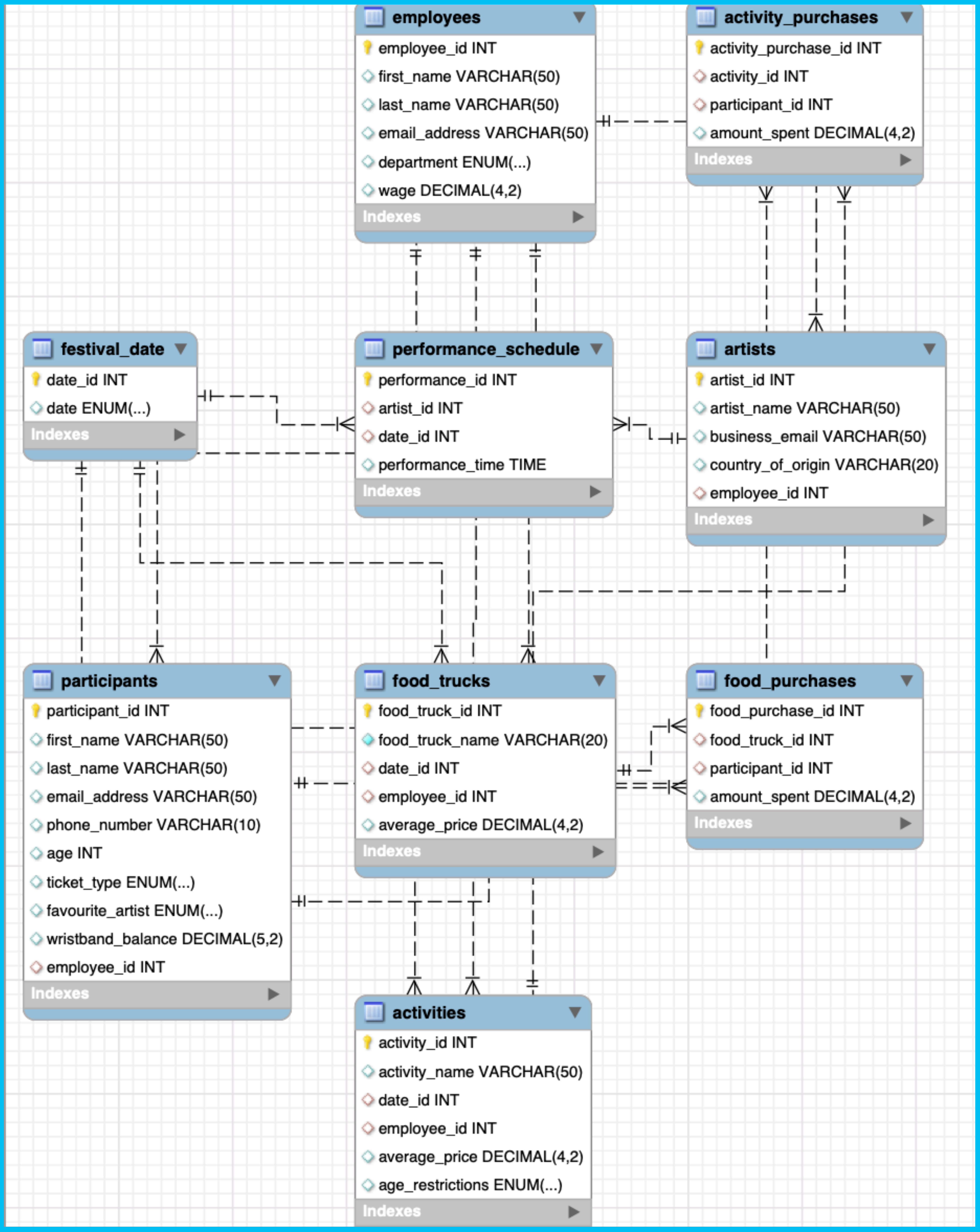
A BRIEF OVERVIEW OF THE SCHEMA...

- the database contains **9 TABLES:**

1. **employees -** information about staff members

2. **participants -** information about people attending the festival + who welcomed each of them to the festival

3. **artists -** artists performing at the festival + who is in charge of them



employees
- employee_id INT
- first_name VARCHAR(50)
- last_name VARCHAR(50)
- email_address VARCHAR(50)
- department ENUM(...)
- wage DECIMAL(4,2)
- Indexes

activity_purchases
- activity_purchase_id INT
- activity_id INT
- participant_id INT
- amount_spent DECIMAL(4,2)
- Indexes

festival_date
- date_id INT
- date ENUM(...)
- Indexes

performance_schedule
- performance_id INT
- artist_id INT
- date_id INT
- performance_time TIME
- Indexes

artists
- artist_id INT
- artist_name VARCHAR(50)
- business_email VARCHAR(50)
- country_of_origin VARCHAR(20)
- employee_id INT
- Indexes

participants
- participant_id INT
- first_name VARCHAR(50)
- last_name VARCHAR(50)
- email_address VARCHAR(50)
- phone_number VARCHAR(10)
- age INT
- ticket_type ENUM(...)
- favourite_artist ENUM(...)
- wristband_balance DECIMAL(5,2)
- employee_id INT
- Indexes

food_trucks
- food_truck_id INT
- food_truck_name VARCHAR(20)
- date_id INT
- employee_id INT
- average_price DECIMAL(4,2)
- Indexes

food_purchases
- food_purchase_id INT
- food_truck_id INT
- participant_id INT
- amount_spent DECIMAL(4,2)
- Indexes

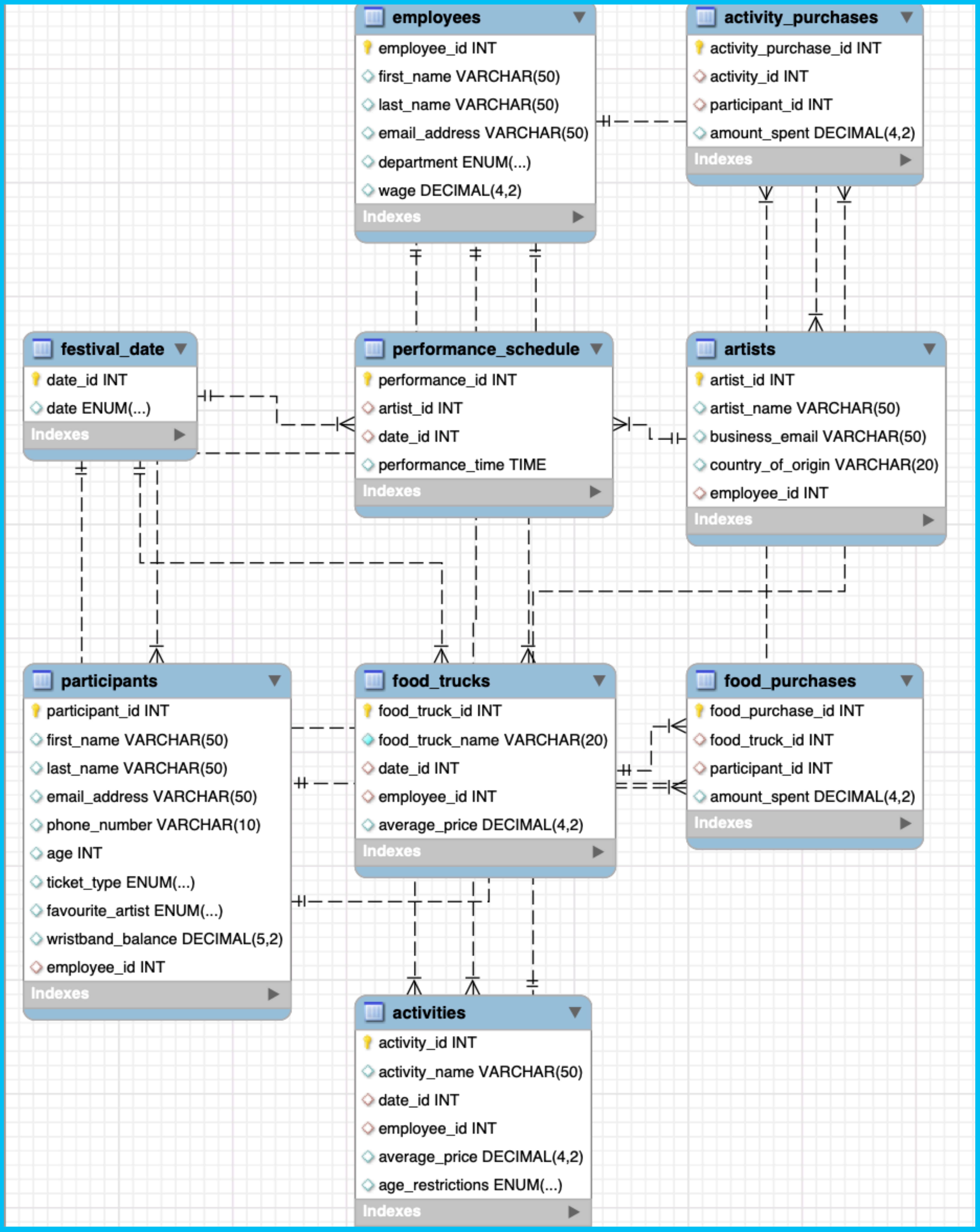
activities
- activity_id INT
- activity_name VARCHAR(50)
- date_id INT
- employee_id INT
- average_price DECIMAL(4,2)
- age_restrictions ENUM(...)
- Indexes

A BRIEF OVERVIEW OF THE SCHEMA...

4. **festival_date -** different dates of the festival

5. **performance_schedule -** the time and day/s the artists are performing

6. **food_trucks -** information about each food truck at the festival



employees
- employee_id INT
- first_name VARCHAR(50)
- last_name VARCHAR(50)
- email_address VARCHAR(50)
- department ENUM(...)
- wage DECIMAL(4,2)
- Indexes

activity_purchases
- activity_purchase_id INT
- activity_id INT
- participant_id INT
- amount_spent DECIMAL(4,2)
- Indexes

festival_date
- date_id INT
- date ENUM(...)
- Indexes

performance_schedule
- performance_id INT
- artist_id INT
- date_id INT
- performance_time TIME
- Indexes

artists
- artist_id INT
- artist_name VARCHAR(50)
- business_email VARCHAR(50)
- country_of_origin VARCHAR(20)
- employee_id INT
- Indexes

participants
- participant_id INT
- first_name VARCHAR(50)
- last_name VARCHAR(50)
- email_address VARCHAR(50)
- phone_number VARCHAR(10)
- age INT
- ticket_type ENUM(...)
- favourite_artist ENUM(...)
- wristband_balance DECIMAL(5,2)
- employee_id INT
- Indexes

food_trucks
- food_truck_id INT
- food_truck_name VARCHAR(20)
- date_id INT
- employee_id INT
- average_price DECIMAL(4,2)
- Indexes

food_purchases
- food_purchase_id INT
- food_truck_id INT
- participant_id INT
- amount_spent DECIMAL(4,2)
- Indexes

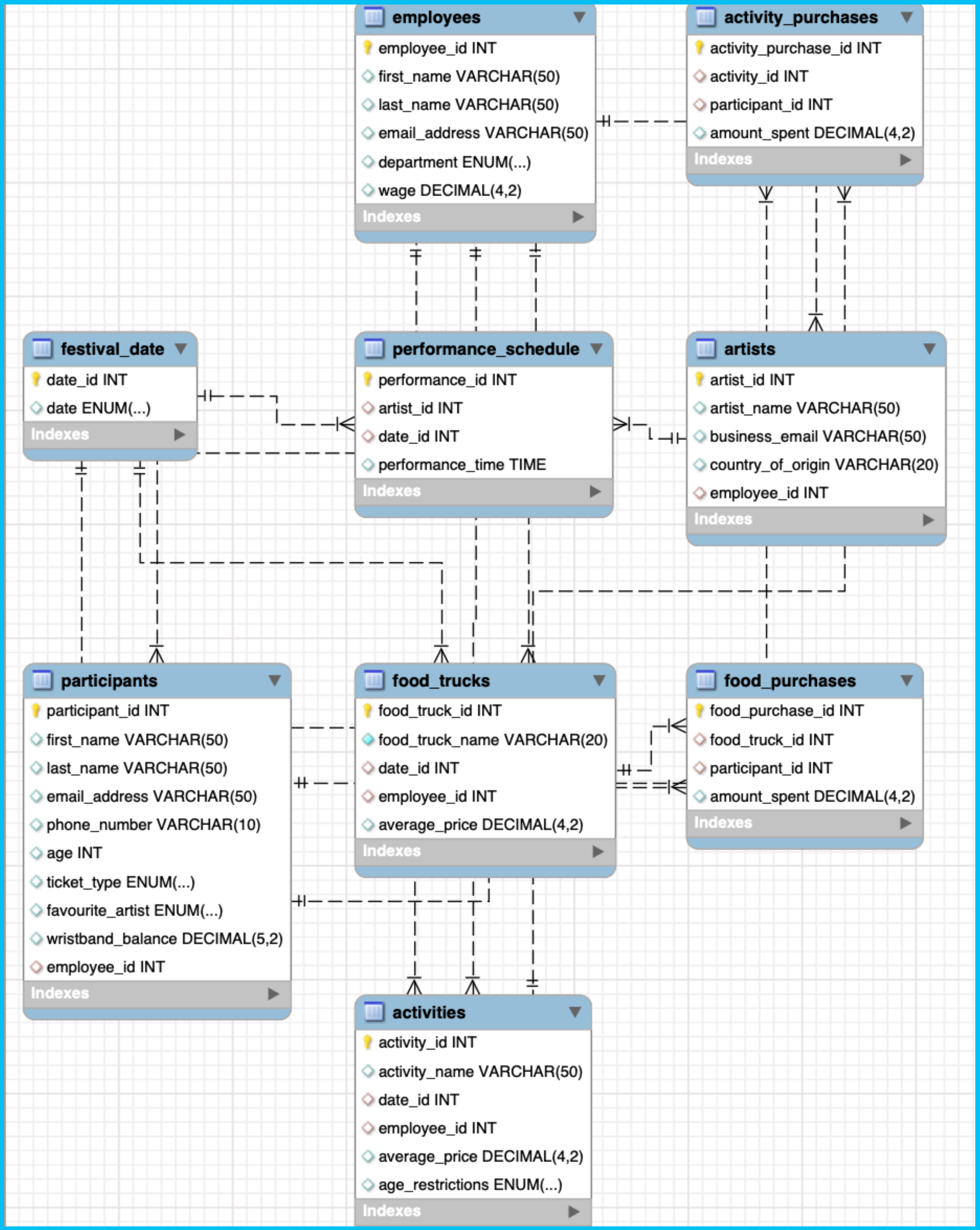
activities
- activity_id INT
- activity_name VARCHAR(50)
- date_id INT
- employee_id INT
- average_price DECIMAL(4,2)
- age_restrictions ENUM(...)
- Indexes

A BRIEF OVERVIEW OF THE SCHEMA...

7. **activities -** information about each activity at the festival

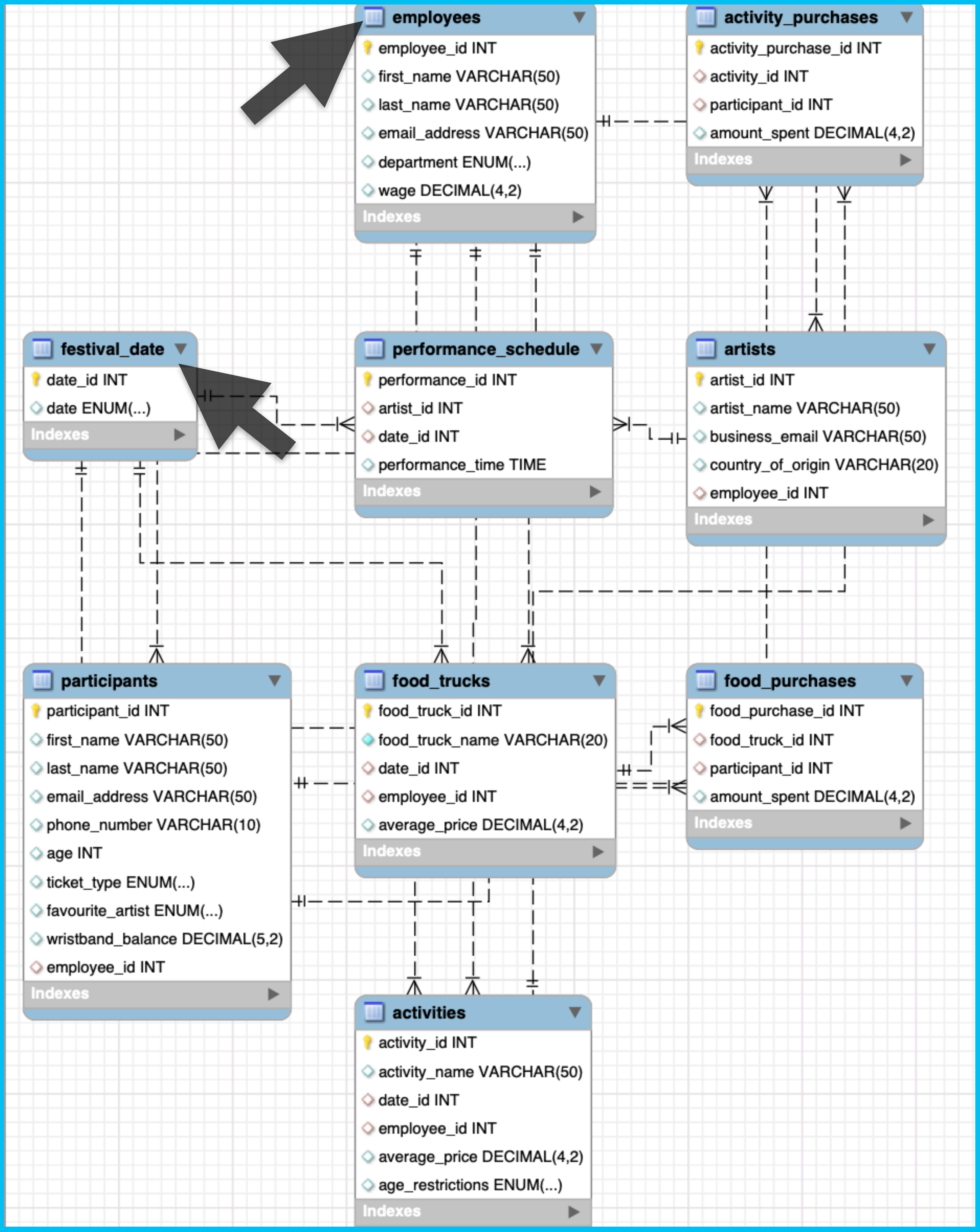
8. **food_purchases -** information about each purchase made at the food trucks

9. **activity_purchases -** information about each purchase made at the activities



A BRIEF OVERVIEW OF THE SCHEMA...

- **2 TABLES** (*employees* and *festival_date*) only have **PRIMARY KEYS** (an INT NOT NULL id column)
- **ALL** the other tables have **PRIMARY KEYS,** as well as **FOREIGN KEYS** - also represented by INT NOT NULL id columns



A BRIEF OVERVIEW OF THE SCHEMA...

employee_id - foreign key in 4 tables

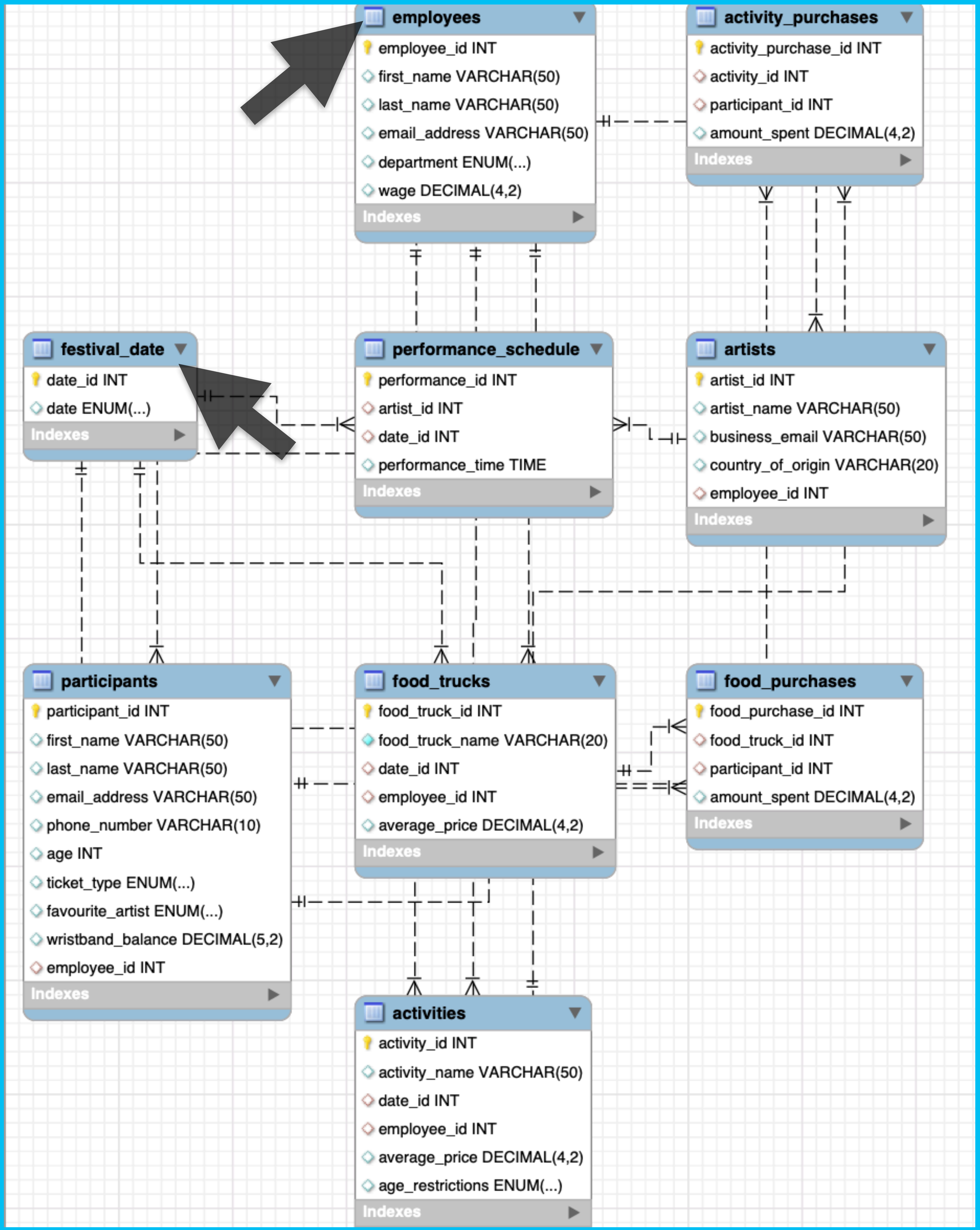
participant_id - foreign key in 2 tables

artist_id - foreign key in 1 table

date_id - foreign key in 3 tables

food_truck_id - foreign key in 1 table

activity_id - foreign key in 1 table



```
employees
  employee_id INT
  first_name VARCHAR(50)
  last_name VARCHAR(50)
  email_address VARCHAR(50)
  department ENUM(...)
  wage DECIMAL(4,2)
  Indexes

activity_purchases
  activity_purchase_id INT
  activity_id INT
  participant_id INT
  amount_spent DECIMAL(4,2)
  Indexes

festival_date
  date_id INT
  date ENUM(...)
  Indexes

performance_schedule
  performance_id INT
  artist_id INT
  date_id INT
  performance_time TIME
  Indexes

artists
  artist_id INT
  artist_name VARCHAR(50)
  business_email VARCHAR(50)
  country_of_origin VARCHAR(20)
  employee_id INT
  Indexes

participants
  participant_id INT
  first_name VARCHAR(50)
  last_name VARCHAR(50)
  email_address VARCHAR(50)
  phone_number VARCHAR(10)
  age INT
  ticket_type ENUM(...)
  favourite_artist ENUM(...)
  wristband_balance DECIMAL(5,2)
  employee_id INT
  Indexes

food_trucks
  food_truck_id INT
  food_truck_name VARCHAR(20)
  date_id INT
  employee_id INT
  average_price DECIMAL(4,2)
  Indexes

food_purchases
  food_purchase_id INT
  food_truck_id INT
  participant_id INT
  amount_spent DECIMAL(4,2)
  Indexes

activities
  activity_id INT
  activity_name VARCHAR(50)
  date_id INT
  employee_id INT
  average_price DECIMAL(4,2)
  age_restrictions ENUM(...)
  Indexes
```

NOW THAT WE COVERED WHAT YOU DON'T SEE... LET'S MOVE ON TO WHAT YOU ACTUALLY SEE!

WHAT CAN YOU DO ON THE APP AS A PARTICIPANT?

WHEN ARE THE ARTISTS PLAYING?

You can see a full concert schedule containing the time and date of each performance, so that you don't miss any of your favourite artists!

```
CREATE VIEW concert_schedule
AS
SELECT a.artist_name, ps.performance_time, d.date
FROM performance_schedule ps
INNER JOIN  artists a
ON a.artist_id = ps.artist_id
INNER JOIN festival_date d
ON d.date_id = ps.date_id
ORDER BY d.date,ps.performance_time;

SELECT * FROM concert_schedule;
```

— by using a view that joins multiple tables and orders the information for an easier reading

WHAT IS YOUR FINAL WRISTBAND BALANCE AT THE END OF THE FESTIVAL?

While you know how much money you initially topped up your electronic wristband with, you will also be able to view your balance at the end of the festival - so you know how much money you need to withdraw!

```sql
SELECT DISTINCT p.participant_id,CONCAT(p.first_name , ' ' , p.last_name) AS
participant, p.wristband_balance, (p.wristband_balance - IFNULL((
                                SELECT SUM(f.amount_spent)
                                FROM food_purchases f
                           WHERE p.participant_id = f.participant_id) -
IFNULL((SELECT SUM(a.amount_spent)
FROM activity_purchases a
WHERE p.participant_id = a.participant_id),0),0)) AS after_purchases
FROM participants p
LEFT JOIN food_purchases f
ON p.participant_id = f.participant_id
ORDER BY p.participant_id;
```

— by using a subquery that adds up the total amount spent on food and activities by each participants, before subtracting it from the original balance

WHAT CAN YOU DO ON THE APP AS A STAFF MEMBER?

YOU CAN SEE...

- how many participants were welcomed to the festival and who these participants are
- who you are working for, if your department has several potential workplaces
- what participants are allowed to do certain activities according to their age
- the total amount of money spent by the participants on food trucks and activities throughout the festival

WHAT PARTICIPANTS CAN DO KAYAKING?

```
DELIMITER //
CREATE FUNCTION agerestrictions(age INT)
RETURNS VARCHAR(3)
READS SQL DATA DETERMINISTIC
BEGIN
DECLARE allowed VARCHAR(3);
IF age >= 18 THEN
SET allowed = 'YES';
ELSEIF age < 18 THEN
SET allowed = 'NO';
END IF;
RETURN allowed;
END //
DELIMITER ;

SELECT p.participant_id, p.first_name, p.last_name, agerestrictions(age) AS
allowed
FROM participants p;
```

— by using a user-defined function that examines the age of each participant and determines whether or not they are over the age of 18 (age restriction specified in the activities table)

FOOD TRUCK OR ACTIVITY DEPARTMENT? WHO ARE YOU ACTUALLY WORKING FOR

```
CREATE VIEW employee_view
AS
SELECT e.first_name, e.last_name, e.department, ft.food_truck_name AS
working_for
FROM food_trucks ft
INNER JOIN employees e
ON e.employee_id = ft.employee_id
UNION
SELECT e.first_name, e.last_name, e.department, act.activity_name
FROM activities act
INNER JOIN employees e
ON e.employee_id = act.employee_id;

SELECT * FROM employee_view;
```

— by using a view that joins and unites multiple tables and shows the name of the food truck or activity each staff member in these departments is working for

THE PHOTOS WE USED ARE FROM:

1. **Summer Well Festival Facebook Page -** *https://www.facebook.com/SummerWellFestival*
2. **Pinterest -** *https://ro.pinterest.com/search/pins/?q=festival&rs=typed&term_meta[]=festival%7Ctyped*

THANK YOU FOR YOUR ATTENTION!

WE HOPE THIS GAVE YOU A LITTLE SENSE OF NORMALITY IN THESE UNUSUAL TIMES, WHILE ALSO MAKING YOU CRAVE A FUN FESTIVAL WITH AWESOME MUSIC!