Planing

Website Explanation

My website will be an index of amusement rides at theme parks or under construction which can then be accessed when looking at a park or a type of ide. Then a more complex feature is a park planner where you input the type of rides that you want to do and then my database will find the park with the most amount of the ride types you were looking for and if i have the time create an option to input your location and calculator which parks are the closest and sort it from there.

Website Layout

Colours:

location	option1	option2	option3	chozen
Background	#edd4b2	#fefae0	#f4f1bb	#fefae0
Secondary background				#f4f1bb
Nav bar colour	#222823	#08090A	#242325	#242325
Drop down colour	#929487	#acadbc	#dbdcda	#7f7f7f

Fonts:

NAVBAR - mediavail sharp

TEXT - lunsio

Footer - Shadows into light two

Header - quintessential

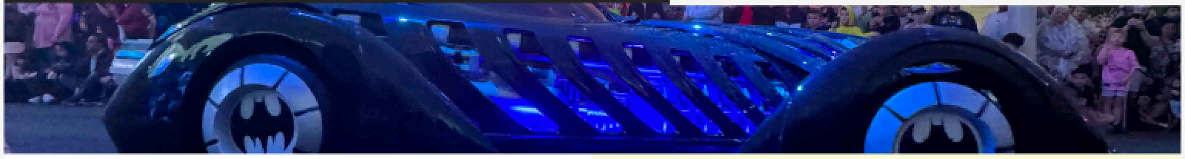
Layout

Page	description
Parks	alternating background with pictures and the name of a theme park, including location. when clicked on it will take you to a list of rides that are in that park. With a simple description included
Rides	alternating background with pictures and the name of the ride, including the manufacture and type. clicking on the ride will pull up more detail and then clicking on that will take you to the parks that have that exact ride or at least the ride type, it will also have a more detailed in depth look at that ride with a full ride layout with the list of elements included
Home	Just a simple parge with a brief description on how to use the website with finding information methods and how to use selection boxes
Manufacturers	A simple page with the list of manufacturers with the same style as the other pages and clicking on a manufacturer will take you to the list of rides they have made sorted by the type
Rides elements	A page containing each ride element like a loop or a corkscrew with a description on the forces the element offers and if in is inverted or not and clicking on the page will take you to a list of rides with that element
Ride types	Found by using the dropdown menu the ride types page is just a list of the ride types using the same format as the manufacturers with a picture of that ride (both the track and train) and the name and a description.

Examples

Flask route /manufacturer

Home Parks Rides manufacturers



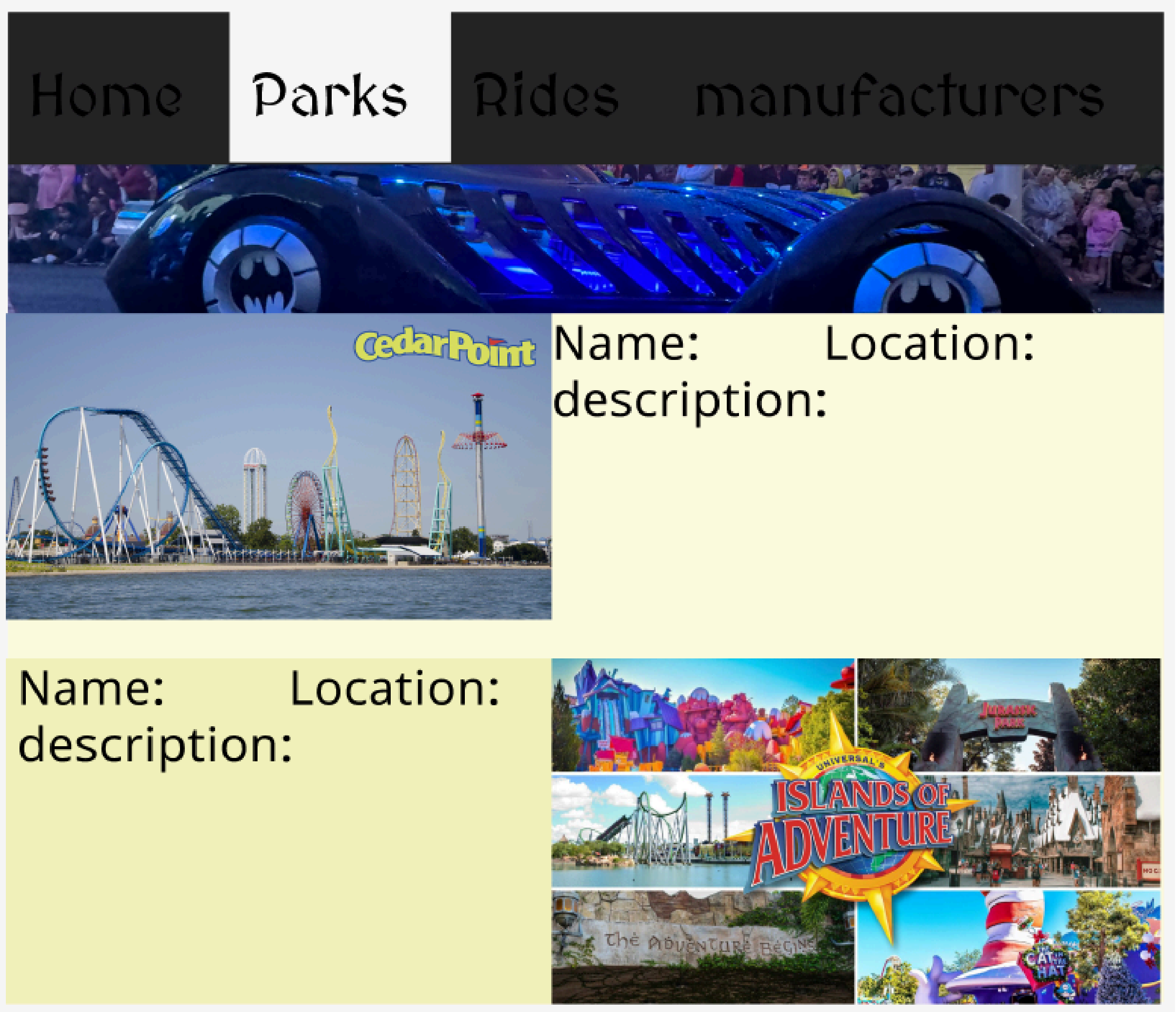
BOLLIGER & MABILLARD

Name: Location: description:

Name: Location: description:

VEKOMA

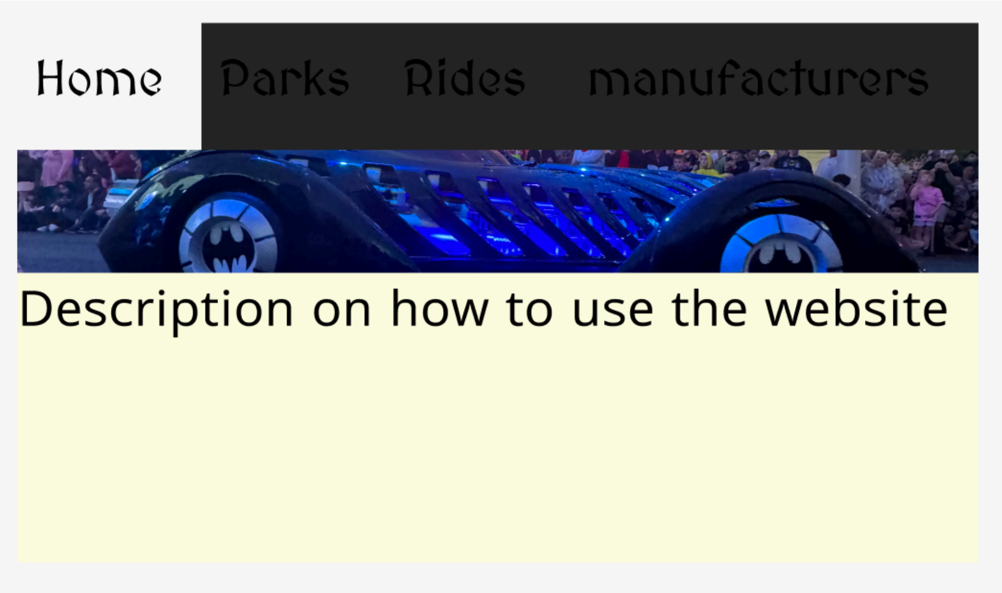
/parks



Name: Location: description:

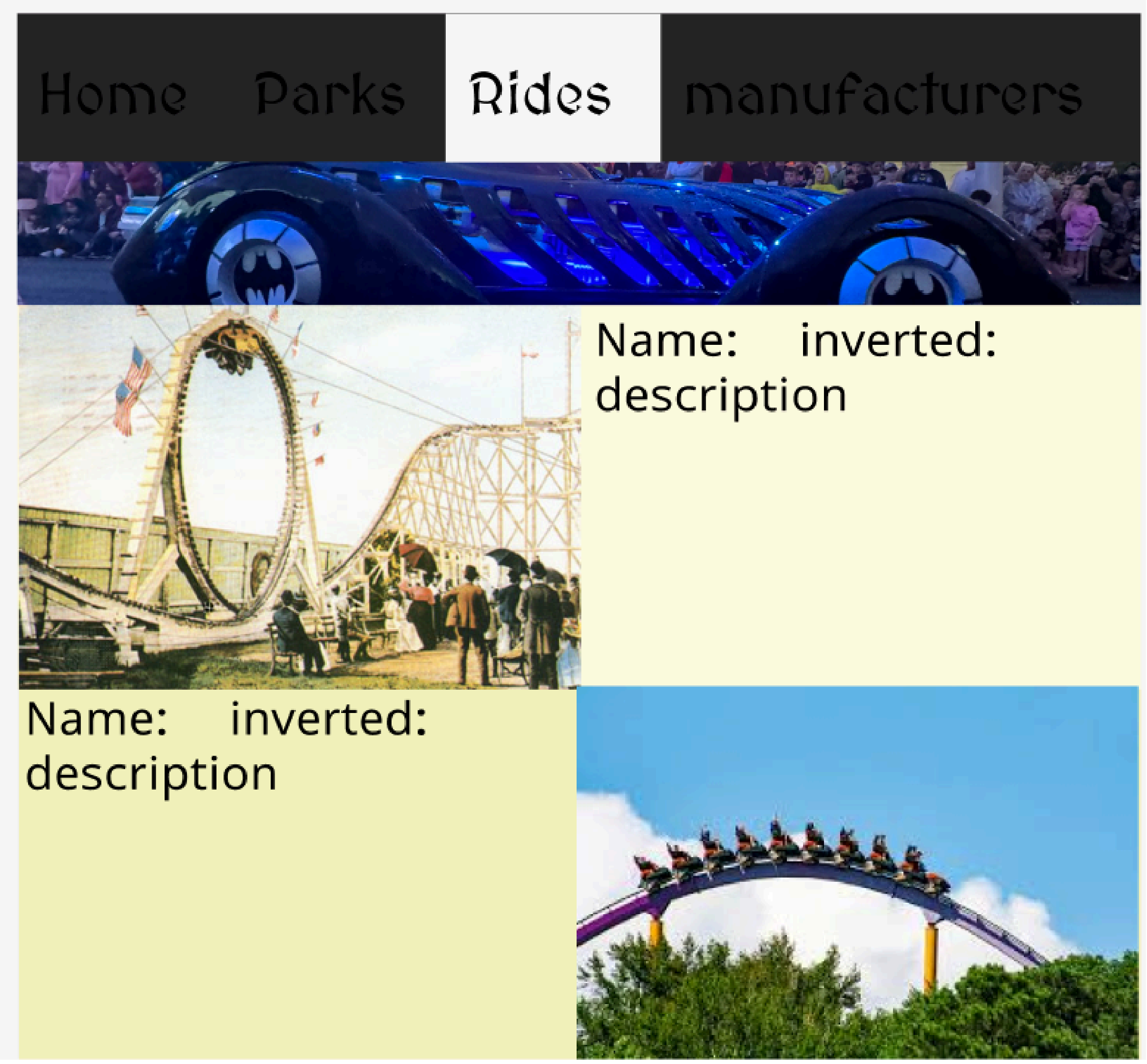
Name: Location: description:

/



Description on how to use the website

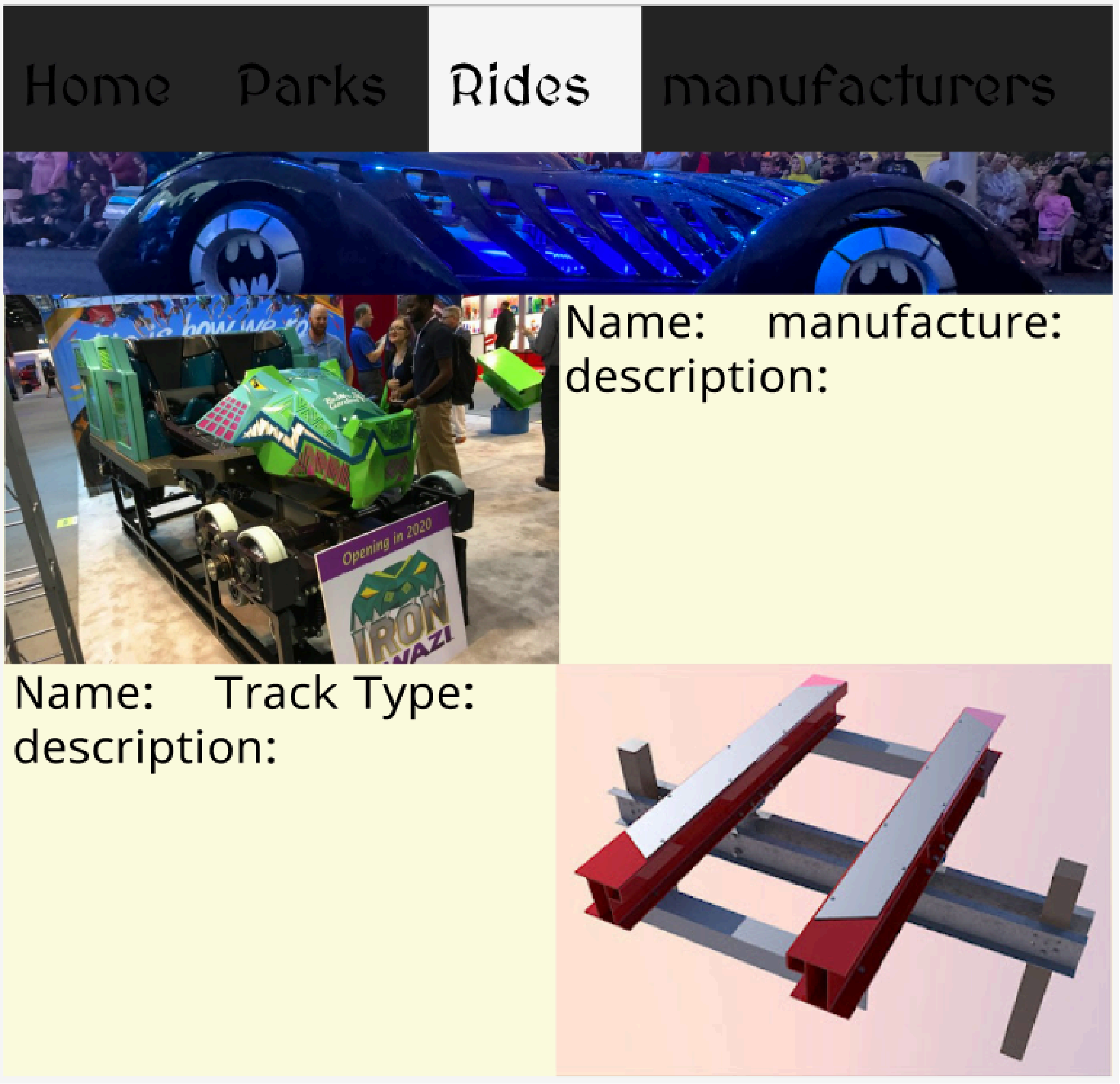
/rideelements



Name: inverted: description

Name: inverted: description

/ridetype



Name: manufacture:
description:

Name: Track Type:
description:

Justification:

My targeted end users of theme park enthusiasts more in terms of rides are a lot more interested in what the actual ride is over all the more atmospheric features of a park which is why the rides have the most detail as that is what ends up bringing everyone to those parks.

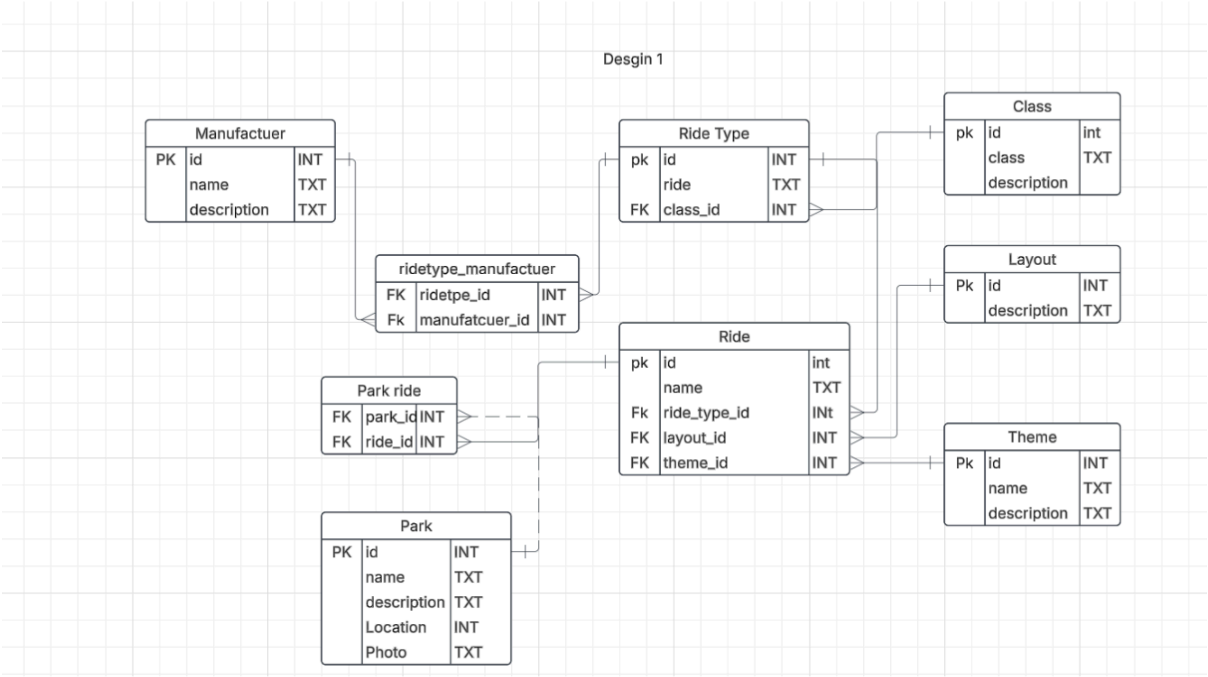
I am simply going to use the navigation by using the name of each page such as rides, parks etc. to make the website simpler to create when clicking on a park taking you to its ride list the url will be for rides but using the id of the park only its rides will be displayed.

Data management

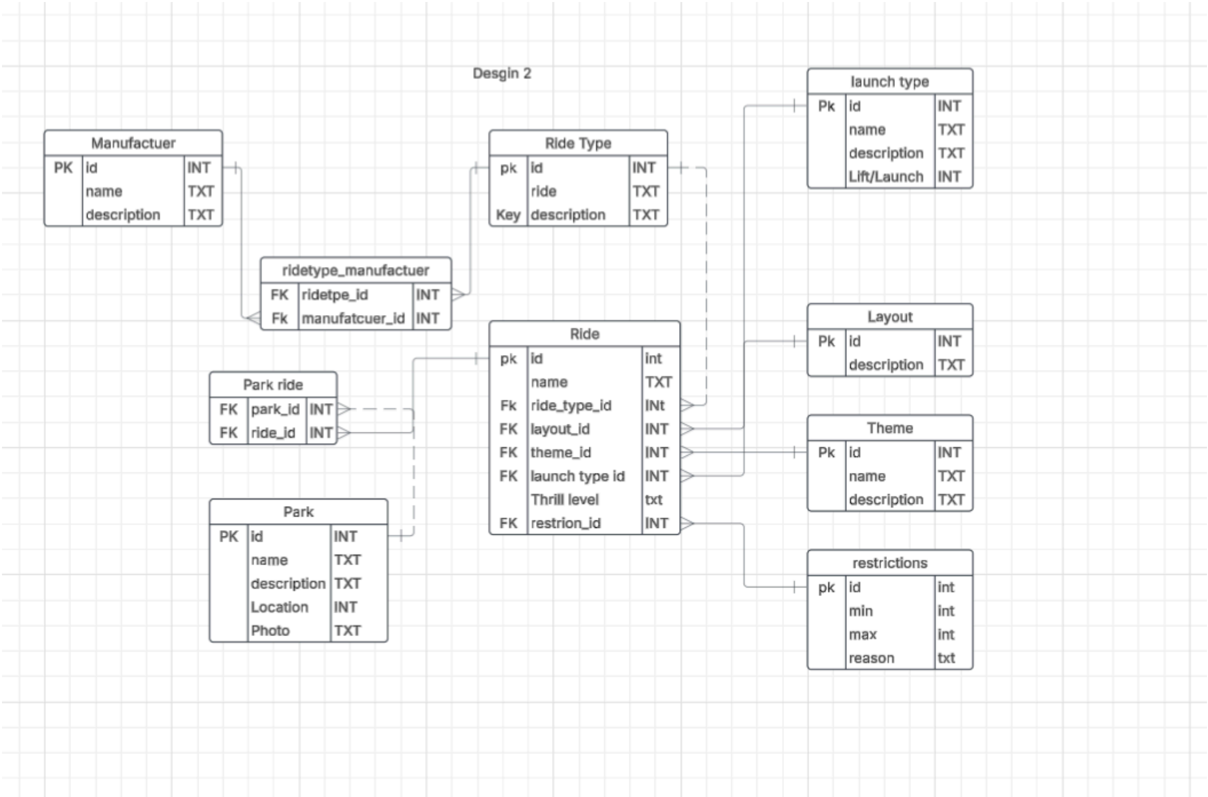
The website will be separated into the main components of the amusement industry being parks, rides and manufacturers. Then in each page it will go through the list of parks with there name and photo and to find out more clicking on the image will take you to that's comments table and give more information about that and links to where it appears elsewhere like when click on a park you can then click on a ride in that park to pull up more information on that because of the complexity of rides and parks without some separation the web page would get too crowded. To find out more on more exact things like just all the ride types or themes then a dropdown menu is used to keep the nav bar looking good without sacrificing detail accessibility, as most of my end users are unlikely to be very tech savvy.

ER Diagrams

Design 1



It Started quite complex as I already new most of the important things that had to be there and what i could separate due to ride layout or even full ride clones existing some even with duplicate names as in the disney parks there are multiple space mountains and big thunder railroads



Desgin 2

Manufactuer

PK	id	INT
	name	TXT
	description	TXT

Ride Type

pk	id	INT
	ride	TXT
Key	description	TXT

launch type

Pk	id	INT
	name	TXT
	description	TXT
	Lift/Launch	INT

ridetype_manufactuer

FK	ridetpe_id	INT
Fk	manufatcuer_id	INT

Ride

pk	id	int
	name	TXT
Fk	ride_type_id	INt
FK	layout_id	INT
FK	theme_id	INT
FK	launch type id	INT
	Thrill level	txt
FK	restrion_id	INT

Layout

Pk	id	INT
	description	TXT

Park ride

FK	park_id	INT
FK	ride_id	INT

Theme

Pk	id	INT
	name	TXT
	description	TXT

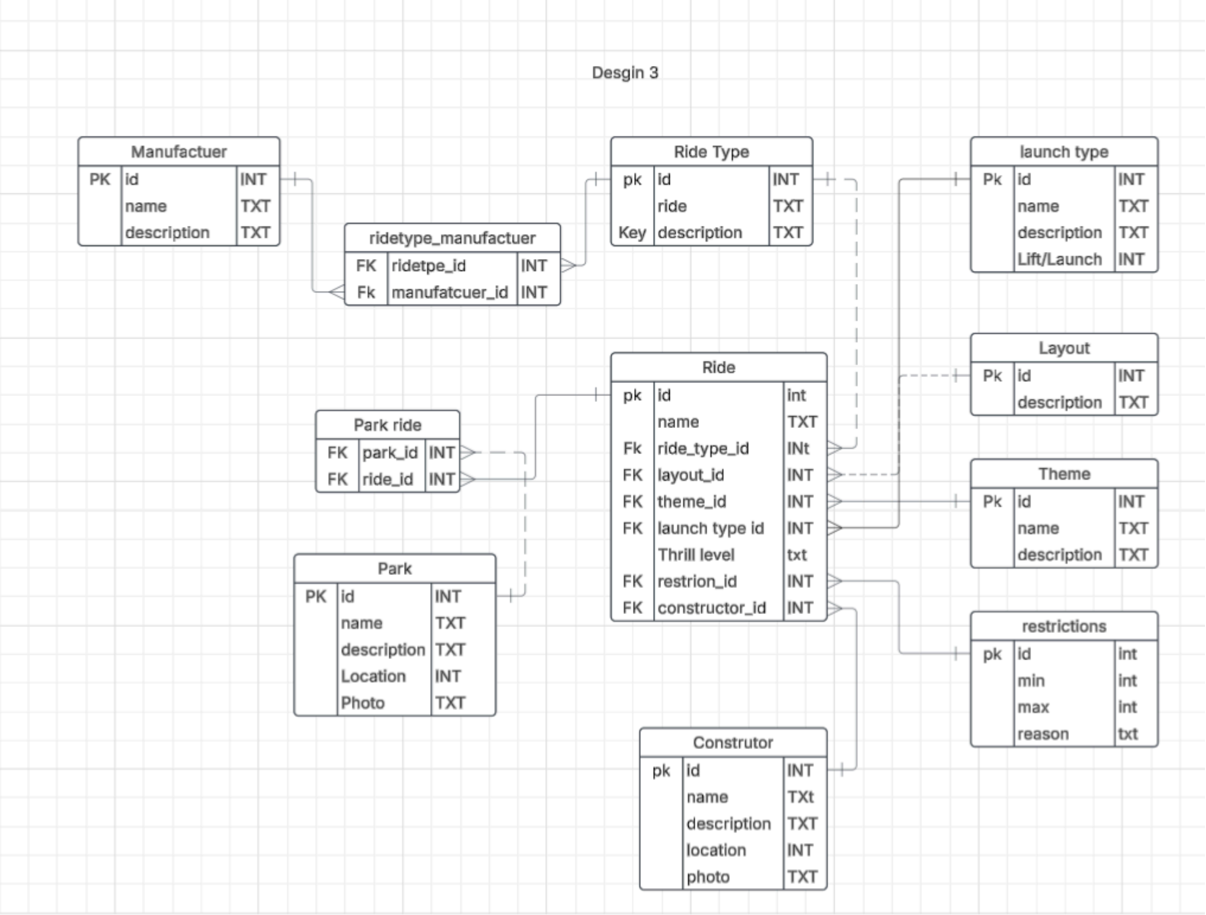
Park

PK	id	INT
	name	TXT
	description	TXT
	Location	INT
	Photo	TXT

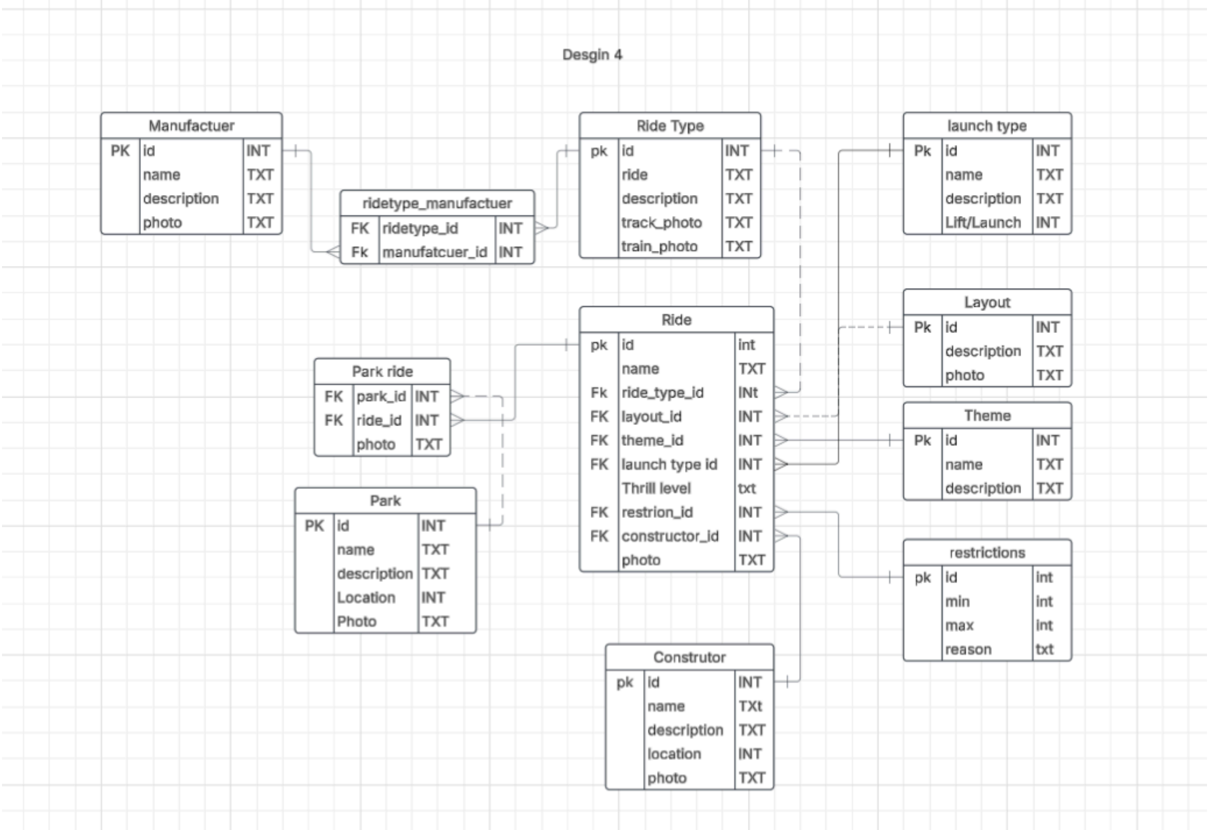
restrictions

pk	id	int
	min	int
	max	int
	reason	txt

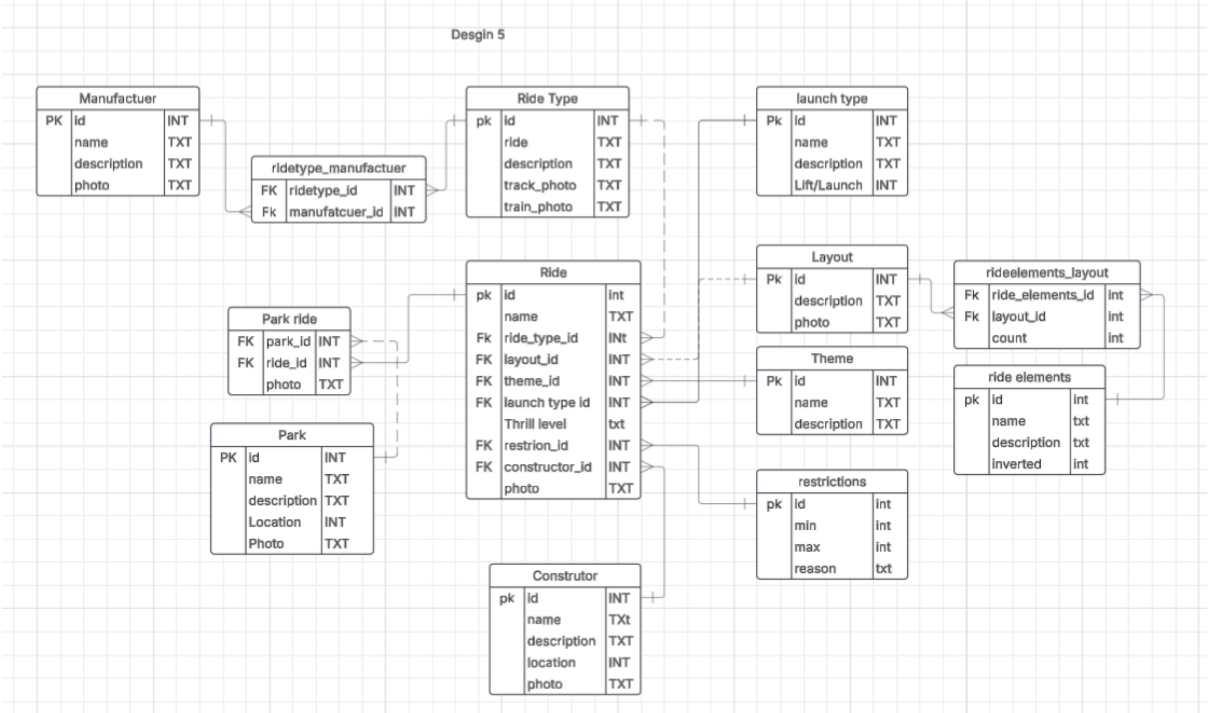
I had to change around some of the tables due to the fact that the main important part is the final ride itself as it can have many changes made from its base type to be more effect for that certain park. I then also added the thrill level which is given to most rides in the world to give a rough estimate to how extreme the ride is. Ride restrictions are also important as if you are too short or tall, or have a medical condition you may not be able to ride some rides.



The constructor table was added as the construction company used to build a ride is different even for the same manufacturer and they can have used different materials and some companies have bad reputations for cost cutting in particular using less bents of wooden roller coasters making them more bumping, jerky and overall a nightmare to maintain.



Extra photos were added where necessary like the photo of the ride in that park and photos for logos and the actual ride type for its track and trains.



To make the layout less exact for each individual ride I can instead give a generic layout description for each ride then have the counter for the more important ride elements like an outer banked turn, a loop or a corkscrew.

SQL/ sqlalchemy

Statements

display rides in a selected park:

```
SELECT
Ride.id, Ride.name, Ride.photo, Ride_Type.ride AS ride_type, Ride.thrill_level
FROM Ride
JOIN Park_Ride ON Ride.id = Park_Ride.ride_id
JOIN Park ON Park_Ride.park_id = Park.id
JOIN Ride_Type ON Ride.ride_type_id = Ride_Type.id
WHERE Park.id = ?
```

Get launched rides:

```
SELECT
Ride.name, Launch_Type.name AS launch_type
FROM Ride
JOIN Launch_Type ON Ride.launch_type_id = Launch_Type.id
WHERE Launch_Type.lift_launch = 1
```

Rides by manufacturer:

```
SELECT Ride.id, Ride.name, Ride.photo
FROM Ride
JOIN RideType_Manufacturer ON Ride.ride_type_id = RideType_Manufacturer.ride_type_id
JOIN Manufacturer ON RideType_Manufacturer.manufacturer_id = Manufacturer.id
WHERE Manufacturer.id = ?
```

Parks ride count:

```
SELECT Park.id, Park.name, Park.photo, COUNT(Park_Ride.ride_id) AS total_rides
FROM Park
LEFT JOIN Park_Ride ON Park.id = Park_Ride.park_id
GROUP BY Park.id, Park.name, Park.photo
ORDER BY total_rides DESC
```

Find all thrill levels for filtering

```
SELECT DISTINCT thrill_level FROM Ride ORDER BY thrill_level ASC
```

Inverting rides

```
SELECT
Ride.id, Ride.name, Ride.photo
FROM Ride
JOIN RideElements_Layout ON Ride.layout_id = RideElements_Layout.layout_id
JOIN Ride_Elements ON RideElements_Layout.ride_elements_id = Ride_Elements.id
WHERE Ride_Elements.inverted = 1
```

Ride details

```
SELECT
Ride.id, Ride.name, Ride.photo,
Ride_Type.ride AS ride_type,
Layout.description AS layout,
Launch_Type.name AS launch_type,
Ride.thrill_level,
Restrictions.min AS min_height, Restrictions.max AS max_height
FROM Ride
JOIN Ride_Type ON Ride.ride_type_id = Ride_Type.id
JOIN Layout ON Ride.layout_id = Layout.id
JOIN Launch_Type ON Ride.launch_type_id = Launch_Type.id
JOIN Restrictions ON Ride.restriction_id = Restrictions.id
WHERE Ride.id = ?
```

Number of rides for each type;

```
SELECT
Ride_Type.ride AS ride_type, COUNT(Ride.id) AS total_rides
FROM Ride
JOIN Ride_Type ON Ride.ride_type_id = Ride_Type.id
GROUP BY Ride_Type.ride
```

Get rides by a park

```
SELECT
Park.name AS park_name, Ride.name AS ride_name
FROM Park_Ride
JOIN Park ON Park_Ride.park_id = Park.id
JOIN Ride ON Park_Ride.ride_id = Ride.id
WHERE Park.name = ?
```

Filter rides by thrill level

```
SELECT id, name, photo FROM Ride
WHERE thrill_level = ?
ORDER BY name ASC
```

Sample outputs are in the views of my database.

Justifications

For the web pages I am going to use selection typing boxes where it will have the direct names of the items in the database sorted alphabetically but as you type it will only show the options that have the same set of starting letters. This approach is used to still give fast search but decrease the amount of spelling that can potentially cause errors. Therefore I do not need to have an id route at the top of the page preventing any errors for undesired id values like 0 or infinity. This benefit is made by the options appearing using id values so it is highly impossible for incorrect values and id errors because the names of the items used for searching come from the id values as this selection box is created by the table. Dropdown menus have a great level of detail whilst not taking up a lot of space combined with the searching options the website is robust having the ability to display all sets of the import and less important information without sacrificing aesthetics. This website is useful to my end users of the theme park enthusiast as there is such a wide variety of parks and rides, with sheer number this website can become a one stop destination to finding the perfect theme park to go to for the rides you are looking for at the shortest possible distance.