Jerome & Gabriel CS340 Group 17 – Project Step 6

URL to our webpage:
http://flip4.engr.oregonstate.edu:9721

Executive Summary

During this project, we received valuable feedback aimed at enhancing the usability and functionality of our website. The Key recommendations included:

- **Stylized Tables:** Following user feedback, we implemented borders around all tables to improve readability, ensuring a smoother user experience.
- **Attribute Naming Conventions:** In response to suggestions, we adjusted attribute names within our database tables for clarity. For example, "in_stock" was changed to "quantity" to better convey the meaning of what we wanted to accomplish.
- **Dropdown Functionality:** As a result of feedback, we addressed concerns about select dropdown functionality. We initially had the drop-down populate the id's of the entities rather than the names of what those id's represented. As a result, we refined the process to allow users to create developers with pre-populated options and introduced text boxes for adding new developers, enhancing user control.
- **NULLable Relationships:** Realized through feedback, we did not implement nullable relationships. As a result, we revised constraints to allow Nullable relationships between Sales and Employees. This allowed smoother data management without the foreign key constraints.
- **Website Structure:** In line with feedback, we redesigned our website to feature seven pages, including a home page and dedicated pages for each entity and intersection table. This helped improve navigational clarity and the flow of the website.
- **Column Header Readability:** Readability was one of the most suggested improvements to our website. We were using header values from our database that did not make the column headers user for users to read. As a result, we adjusted column headers across the website to use more intuitive titles, such as "ID", "First Name," "Last Name," "Phone", and "Hire Date." This was changed for the Employees webpage. We adopted this style throughout the other web pages.

These enhancements have significantly improved the usability and user experience of our website. Because of this, we are now able to facilitate smoother data management for the user.

Retro Games

By Jerome Cagado and Gabriel-Jean Bertrand

Overview

Retro Games is a unique gaming retailer exclusively offering physical versions of video games in the Seattle area. The business has experienced a surge in sales, reaching $250,000 in the past year. In response to the video game market now favoring digital copies over physical copies, Retro Games needs assistance in a database system to keep up with the growing demand.

The database-driven website will track Retro Games inventory, employees, customers, and sales. By implementing this database, Retro Games will enhance their operational efficiency, ensuring accurate inventory control.

Database Outline

Employees: records the details of Employees who work for Retro Games.

- employee_id: INT, auto_increment, unique, not NULL, PK
- employee_fname: VARCHAR(45), not NULL
- employee_lname: VARCHAR(45), not NULL
- employee_phone: INT, not NULL
- hire_date: DATE, YYYY/MM/DD
- Relationship:
  - A 1:M relationship between **Employees** and **Sales** with employee_id as an FK in **Sales.**

Sales: records details of video games sold.

- sale_id: INT, auto_increment, unique, not NULL, PK
- employee_id: INT, not NULL, FK
- customer_id: INT, not NULL, FK
- sale_revenue: DECIMAL, not NULL
- sold_date: DATE, YYYY/MM/DD
- Relationship:

    - A M: N relationship between **VideoGames** and Sales. Utilizes an intersection table called **VideoGameSales** with sale_id as FK.

VideoGames: records details of video games in Retro Games inventory.

- video_game_id: INT, auto_increment, not NULL, PK
- developer_id: INT, not NULL FK
- video_game_name: VARCHAR(45), not NULL
- price: DECIMAL, not NULL
- quantity: INT, not NULL
- Relationship:
    - A M:N relationship between **VideoGames** and **Sales**. Utilizes an intersection table called **VideoGameSales** with video_game_id as FK.

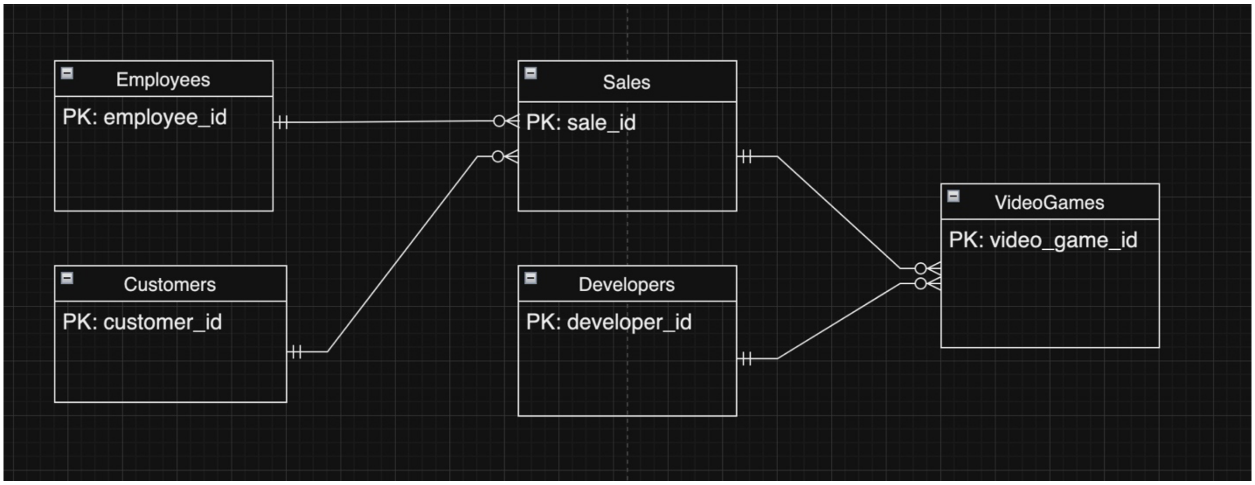
Developers: records details of video game developers.

- developer_id: INT, auto_increment, not NULL PK
- developer_name: VARCHAR(45), not NULL
- developer_country: VARCHAR(45), not NULL
- developer_email: VARCHAR(45), not NULL
- Relationship:
    - A 1:M relationship between **Developers** and **VideoGames** with developer_id as FK.

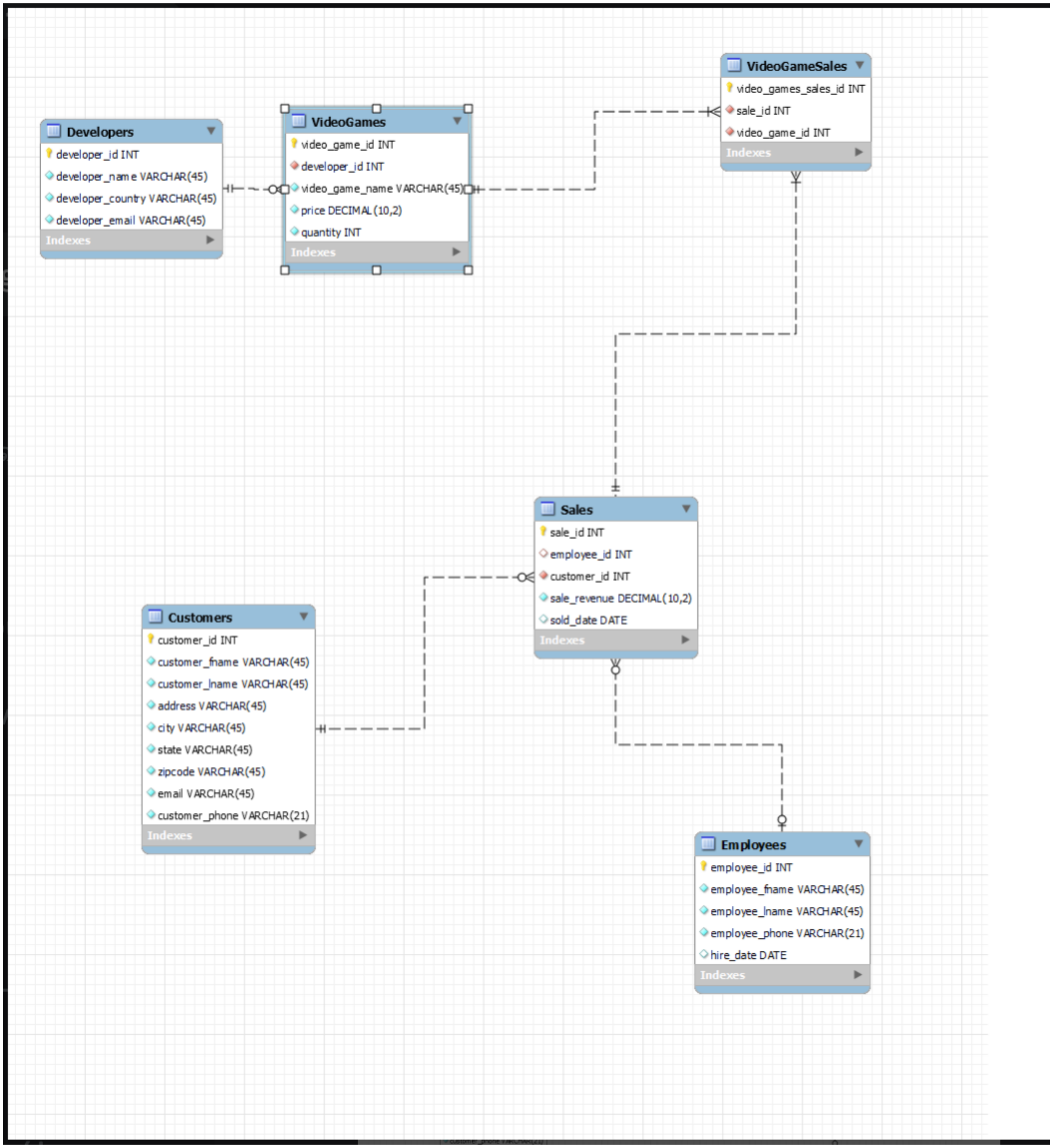
Customers: records details of customers of Retro Games.

- customer_id: INT, auto_increment, not NULL, PK
- customer_fname: VARCHAR(45), not NULL
- customer_lname: VARCHAR(45), not NULL
- address: VARCHAR(45), not NULL
- city: VARCHAR(45), not NULL
- state: VARCHAR(45), not NULL
- zipcode: VARCHAR(45), not NULL
- email: VARCHAR(45), not NULL
- customer_phone: INT, not NULL
- Relationship:
    - A 1:M relationship between **Customers** and **Sales** with customer_id as a FK in **Sales.**

Entity-Relationship Diagram



SCHEMA



Developers

developer_id INT
developer_name VARCHAR(45)
developer_country VARCHAR(45)
developer_email VARCHAR(45)
Indexes

VideoGames

video_game_id INT
developer_id INT
video_game_name VARCHAR(45)
price DECIMAL(10,2)
quantity INT
Indexes

VideoGameSales

video_games_sales_id INT
sale_id INT
video_game_id INT
Indexes

Sales

sale_id INT
employee_id INT
customer_id INT
sale_revenue DECIMAL(10,2)
sold_date DATE
Indexes

Customers

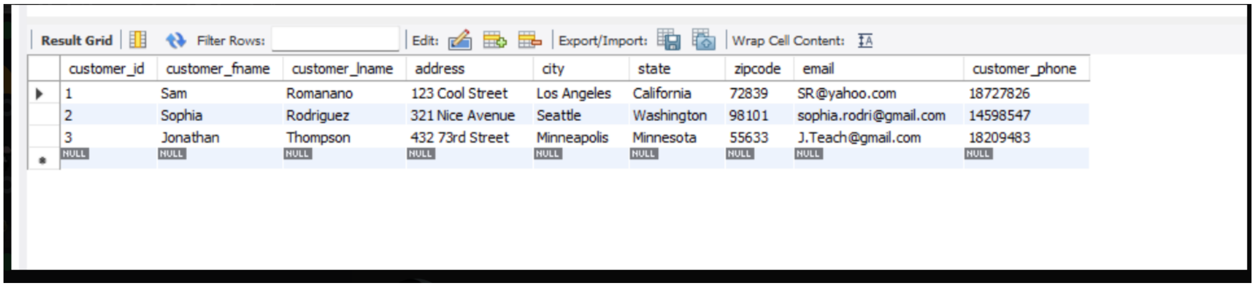
customer_id INT
customer_fname VARCHAR(45)
customer_lname VARCHAR(45)
address VARCHAR(45)
city VARCHAR(45)
state VARCHAR(45)
zipcode VARCHAR(45)
email VARCHAR(45)
customer_phone VARCHAR(21)
Indexes

Employees

employee_id INT
employee_fname VARCHAR(45)
employee_lname VARCHAR(45)
employee_phone VARCHAR(21)
hire_date DATE
Indexes

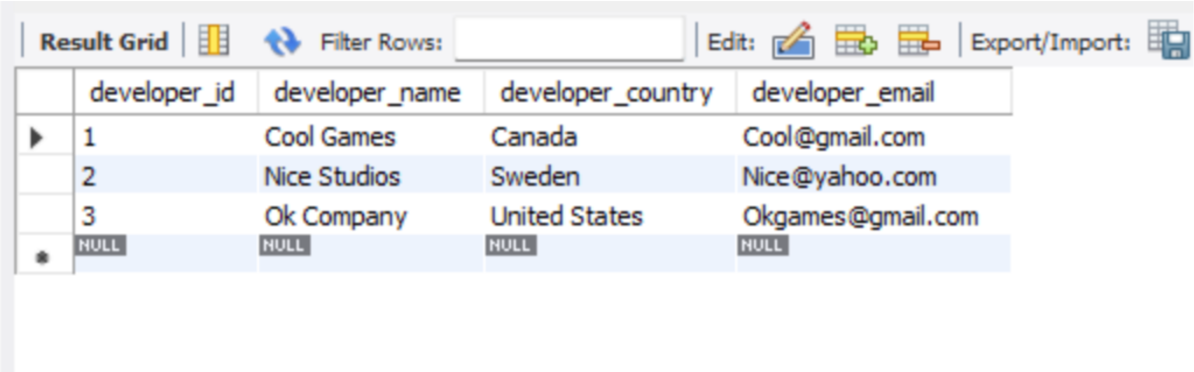
SAMPLE DATA

Customers



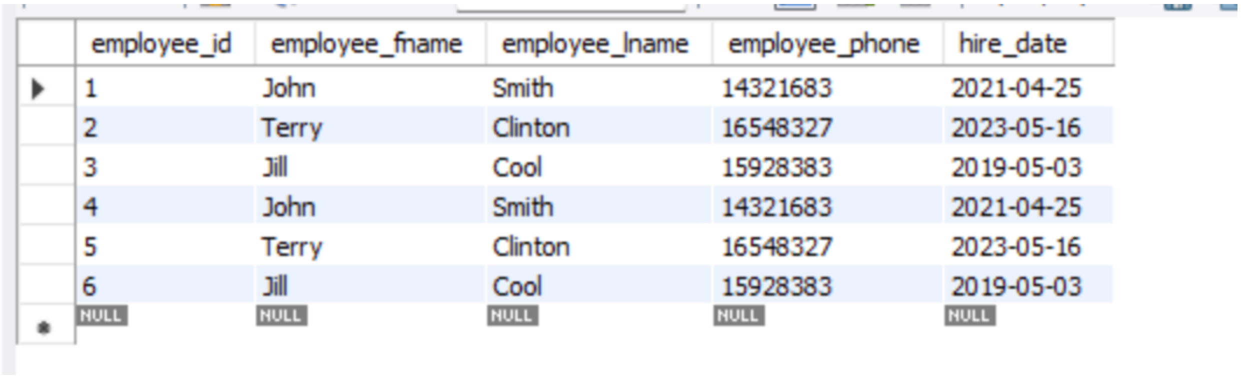
customer_id	customer_fname	customer_lname	address	city	state	zipcode	email	customer_phone
1	Sam	Romanano	123 Cool Street	Los Angeles	California	72839	SR@yahoo.com	18727826
2	Sophia	Rodriguez	321 Nice Avenue	Seattle	Washington	98101	sophia.rodri@gmail.com	14598547
3	Jonathan	Thompson	432 73rd Street	Minneapolis	Minnesota	55633	J.Teach@gmail.com	18209483
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Developers



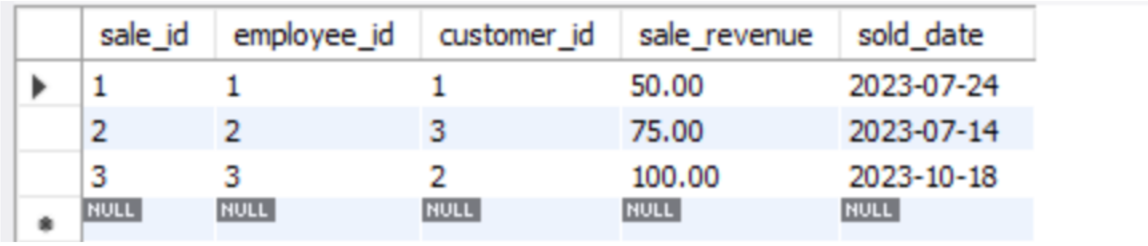
developer_id	developer_name	developer_country	developer_email
1	Cool Games	Canada	Cool@gmail.com
2	Nice Studios	Sweden	Nice@yahoo.com
3	Ok Company	United States	Okgames@gmail.com
NULL	NULL	NULL	NULL

Employees



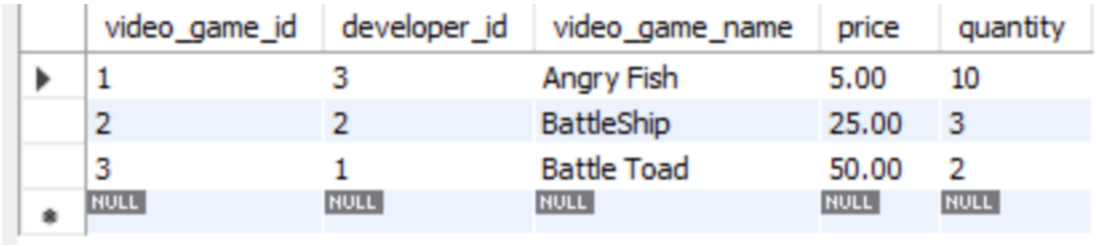
employee_id	employee_fname	employee_lname	employee_phone	hire_date
1	John	Smith	14321683	2021-04-25
2	Terry	Clinton	16548327	2023-05-16
3	Jill	Cool	15928383	2019-05-03
4	John	Smith	14321683	2021-04-25
5	Terry	Clinton	16548327	2023-05-16
6	Jill	Cool	15928383	2019-05-03
NULL	NULL	NULL	NULL	NULL

Sales



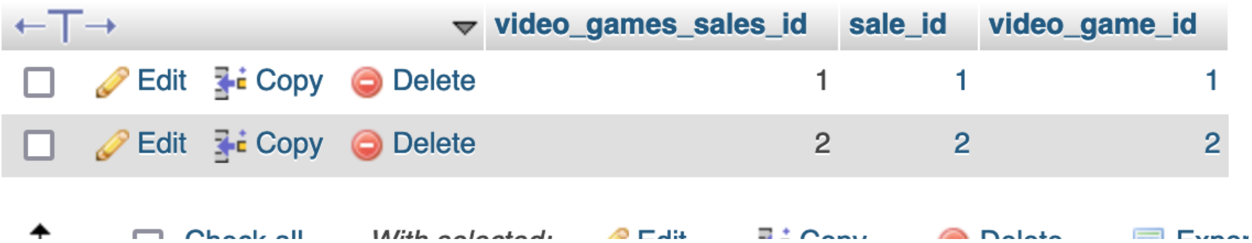
sale_id	employee_id	customer_id	sale_revenue	sold_date
1	1	1	50.00	2023-07-24
2	2	3	75.00	2023-07-14
3	3	2	100.00	2023-10-18
NULL	NULL	NULL	NULL	NULL

VideoGames



video_game_id	developer_id	video_game_name	price	quantity
1	3	Angry Fish	5.00	10
2	2	BattleShip	25.00	3
3	1	Battle Toad	50.00	2
NULL	NULL	NULL	NULL	NULL

VideoGamesSales

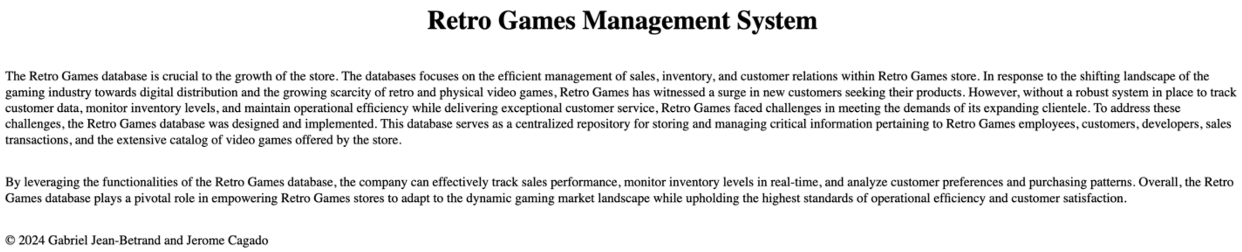


	video_games_sales_id	sale_id	video_game_id
□ Edit Copy Delete	1	1	1
□ Edit Copy Delete	2	2	2

UI SCREENSHOTS:

HOME PAGE:

This homepage provides an intro to the database and links to all entities and the intersection table.



Retro Games Management System

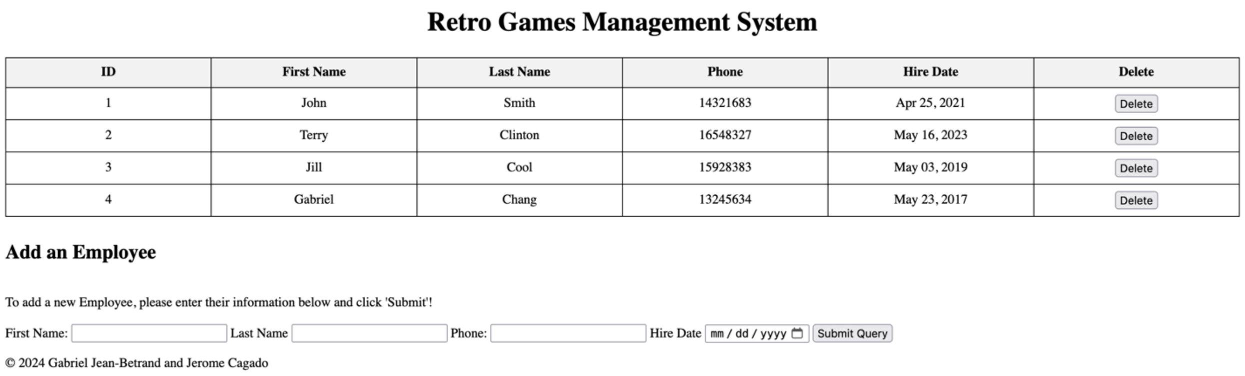
The Retro Games database is crucial to the growth of the store. The databases focuses on the efficient management of sales, inventory, and customer relations within Retro Games store. In response to the shifting landscape of the gaming industry towards digital distribution and the growing scarcity of retro and physical video games, Retro Games has witnessed a surge in new customers seeking their products. However, without a robust system in place to track customer data, monitor inventory levels, and maintain operational efficiency while delivering exceptional customer service, Retro Games faced challenges in meeting the demands of its expanding clientele. To address these challenges, the Retro Games database was designed and implemented. This database serves as a centralized repository for storing and managing critical information pertaining to Retro Games employees, customers, developers, sales transactions, and the extensive catalog of video games offered by the store.

By leveraging the functionalities of the Retro Games database, the company can effectively track sales performance, monitor inventory levels in real-time, and analyze customer preferences and purchasing patterns. Overall, the Retro Games database plays a pivotal role in empowering Retro Games stores to adapt to the dynamic gaming market landscape while upholding the highest standards of operational efficiency and customer satisfaction.

© 2024 Gabriel Jean-Betrand and Jerome Cagado

Employees(CREATE/INSERT/ADD/DELETE)

Allows the user to read and create a new employee. By clicking on delete, this will remove an employee dynamically. Employee is a FK for Sales, and we enabled Sales to be Nullable along with employee to allow deletion of an employee even if they were to sell a videogame.



Retro Games Management System

ID	First Name	Last Name	Phone	Hire Date	Delete
1	John	Smith	14321683	Apr 25, 2021	Delete
2	Terry	Clinton	16548327	May 16, 2023	Delete
3	Jill	Cool	15928383	May 03, 2019	Delete
4	Gabriel	Chang	13245634	May 23, 2017	Delete

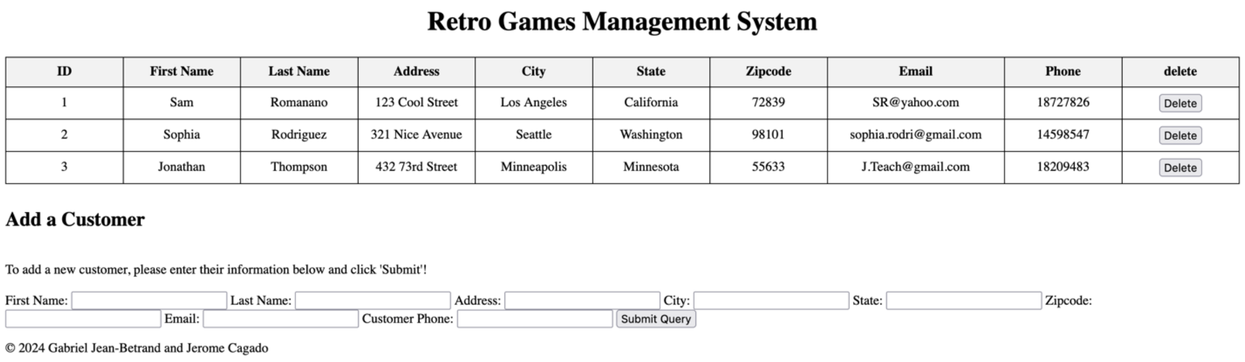
Add an Employee

To add a new Employee, please enter their information below and click 'Submit'!

© 2024 Gabriel Jean-Betrand and Jerome Cagado

Customers(CREATE/INSERT/ADD/DELETE)

Allows the user to read and create a new customer. By clicking on delete, this will remove the customer dynamically.



Retro Games Management System

ID	First Name	Last Name	Address	City	State	Zipcode	Email	Phone	delete
1	Sam	Romanano	123 Cool Street	Los Angeles	California	72839	SR@yahoo.com	18727826	Delete
2	Sophia	Rodriguez	321 Nice Avenue	Seattle	Washington	98101	sophia.rodri@gmail.com	14598547	Delete
3	Jonathan	Thompson	432 73rd Street	Minneapolis	Minnesota	55633	J.Teach@gmail.com	18209483	Delete

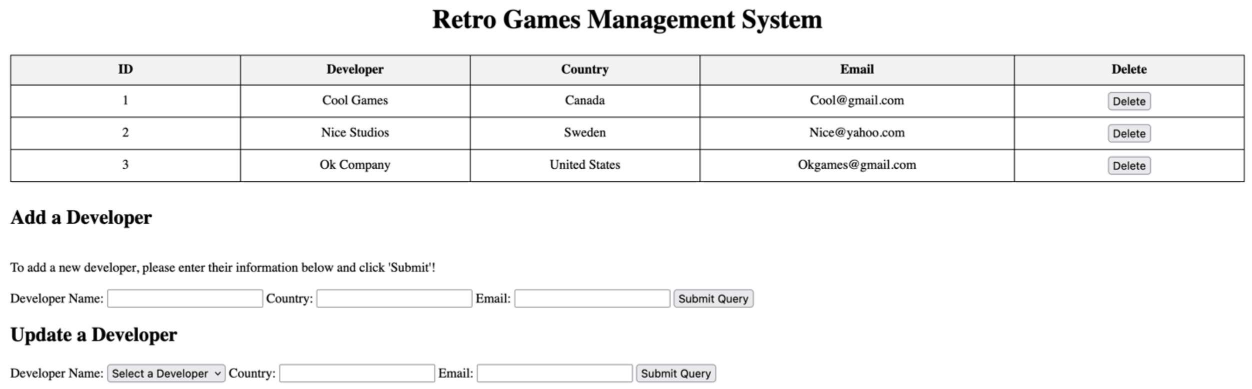
Add a Customer

To add a new customer, please enter their information below and click 'Submit'!

© 2024 Gabriel Jean-Betrand and Jerome Cagado

Developers(CREATE/UPDATE/ADD/DELETE)

Allows user to read and add a new developer. By clicking on delete, this will dynamically remove a developer. In addition, this allows the user to update a developer using a dynamically populated drop-down of current developers. By selecting a developer you want to update, you can edit their country, and email. This will dynamically edit/update the developer in the table.



Retro Games Management System

ID	Developer	Country	Email	Delete
1	Cool Games	Canada	Cool@gmail.com	Delete
2	Nice Studios	Sweden	Nice@yahoo.com	Delete
3	Ok Company	United States	Okgames@gmail.com	Delete

Add a Developer

To add a new developer, please enter their information below and click 'Submit'!

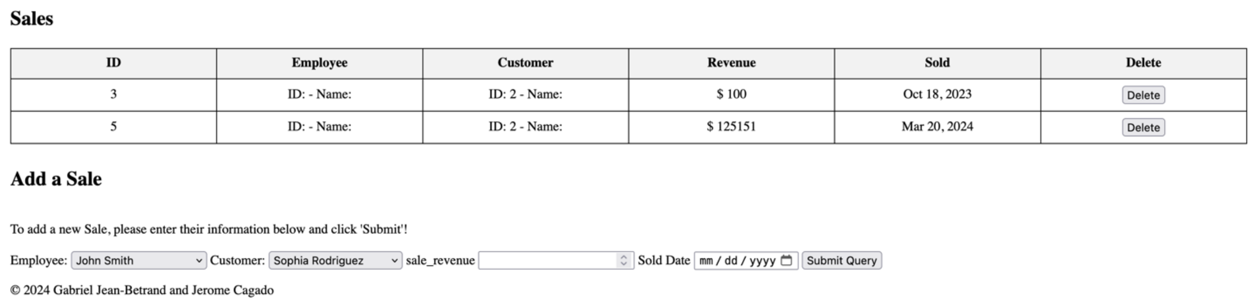
Developer Name: Country: Email: Submit Query

Update a Developer

Developer Name: Select a Developer Country: Email: Submit Query

Sales(CREATE/ADD/DELETE)

This allows the user to read and create a new sale by using the dynamically populated drop-down menu. To add a sale, you must add an employee by using the drop-down menu, and a customer by using the drop-down menu. Sales can be nullable and can be deleted if delete is selected. This table will also show an employee to also be nullable.



Sales

ID	Employee	Customer	Revenue	Sold	Delete
3	ID: - Name:	ID: 2 - Name:	$ 100	Oct 18, 2023	Delete
5	ID: - Name:	ID: 2 - Name:	$ 125151	Mar 20, 2024	Delete

Add a Sale

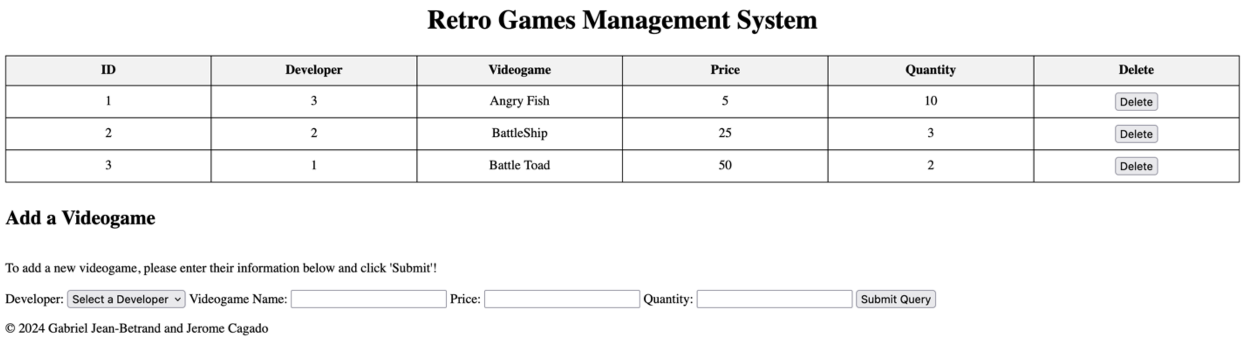
To add a new Sale, please enter their information below and click 'Submit'!

Employee: John Smith Customer: Sophia Rodriguez sale_revenue Sold Date mm / dd / yyyy Submit Query

© 2024 Gabriel Jean-Betrand and Jerome Cagado

Videogame(CREATE/ADD/DELETE)

This allows the user to read and add a new videogame by dynamically picking a developer of said videogame you want to add. By clicking on delete, this will dynamically delete the videogame from the table.



Retro Games Management System

ID	Developer	Videogame	Price	Quantity	Delete
1	3	Angry Fish	5	10	Delete
2	2	BattleShip	25	3	Delete
3	1	Battle Toad	50	2	Delete

Add a Videogame

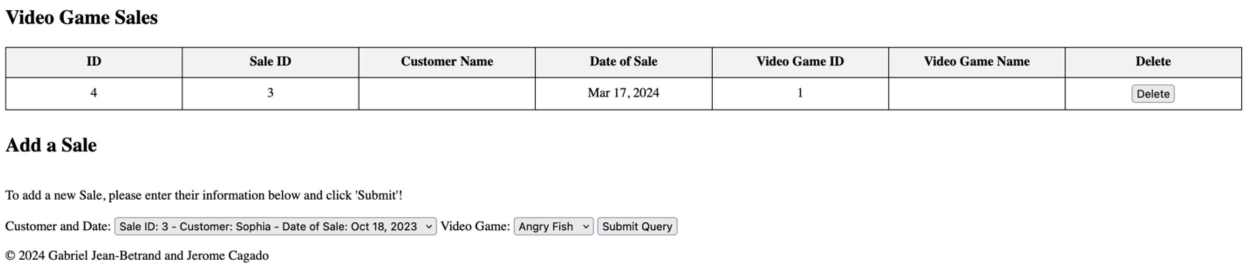
To add a new videogame, please enter their information below and click 'Submit'!

Developer: Select a Developer Videogame Name: Price: Quantity: Submit Query

© 2024 Gabriel Jean-Betrand and Jerome Cagado

VideoGameSales(CREATE/ADD/DELETE)

This allows the user to read and add a new sale to VideoGameSales. To get the sale_id of the sale you must add a customer and the date it was sold by using the pre-populated drown down menu. You must also select a video game from the drop-down menu to create a new VideoGameSales item. If you click on delete, this will delete the videogame sale from this table. This table can have the customer, and videogame be Nullable. In addition, if the sale were deleted from the Sales table, it would be removed from this table as well.



Video Game Sales

ID	Sale ID	Customer Name	Date of Sale	Video Game ID	Video Game Name	Delete
4	3		Mar 17, 2024	1		Delete

Add a Sale

To add a new Sale, please enter their information below and click 'Submit'!

Customer and Date: Sale ID: 3 - Customer: Sophia - Date of Sale: Oct 18, 2023 Video Game: Angry Fish Submit Query

© 2024 Gabriel Jean-Betrand and Jerome Cagado