Pharos University in Alexandria

Data Base Project Documentation

Youssef Elsayed Abdelkader 201900314

Ibrahim Elsayed Abdelkader 201901434

Presented To Eng. Veronia Zaki

12/31/2022

Project Description

The "mechanical" aspect of inventory management is ordering systems. These are the software applications that convert our projections, actual orders, and order amounts into purchase orders or production orders. Hence, we've built a scalable food ordering system calling it

"Instaorder" having a fair amount of **functionalities** as

- Ordering
- Billing
- Security (i.e. User Authentication)

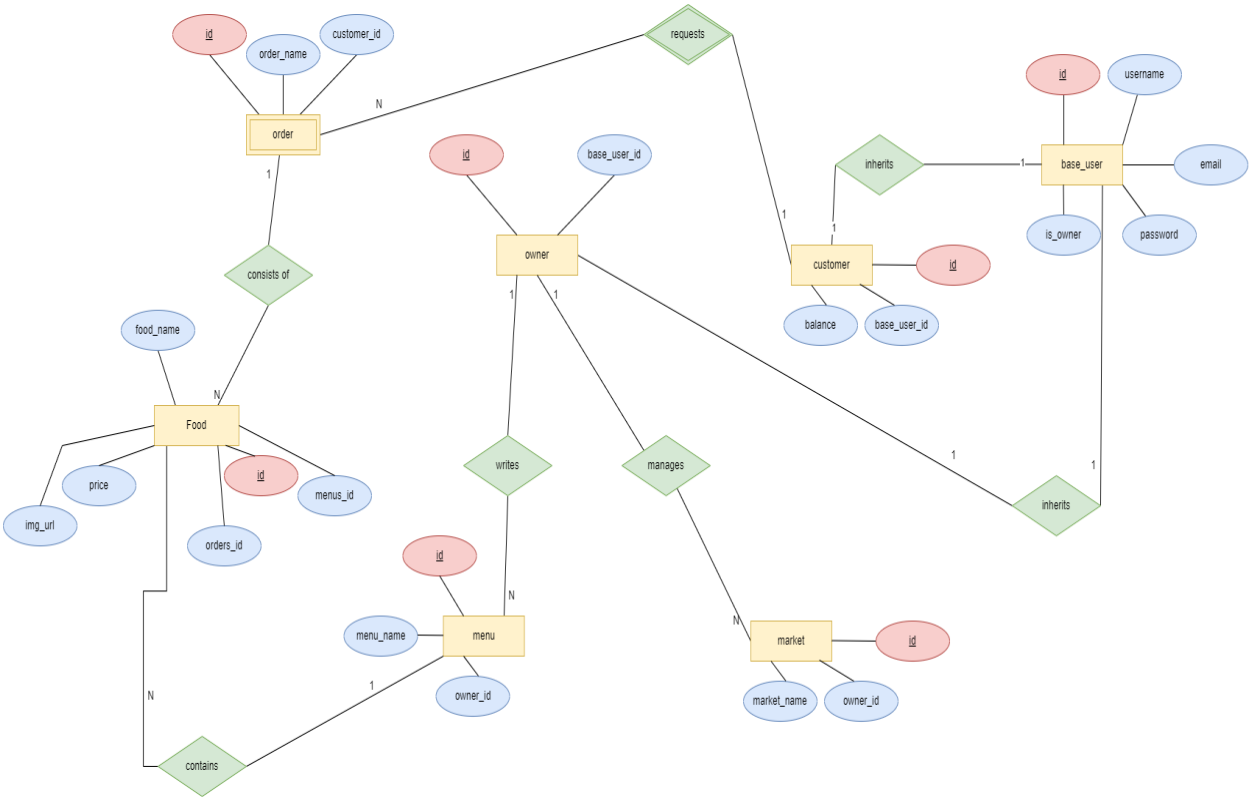
End Users Defined as:

- Employee
- Students

Project Idea: We just make it up.

System Design:

ERD:



General Info about Tools used in the build process:

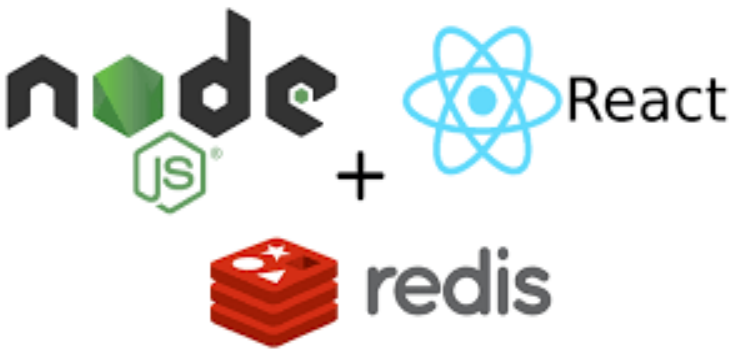
Front End: React JS, Tailwind

Back End: Node JS, Express

Front End Back End Communication: Restful API using tRPC

Database: SQL Server

NOsql (Bonus Part): REDIS -> for storing users sessions



General Implementation info and used queries:

- Joins

- Nested Queries
- Usage of advanced Queries (i.e. grouping)

- Use Index and Views & Built-in functions (Unique Index on order_name column in orders table)

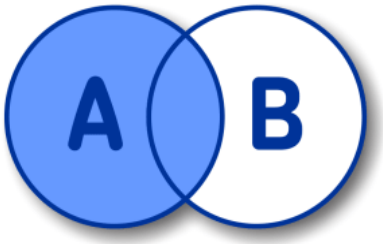
- Creation with an insertion.

- Edit.

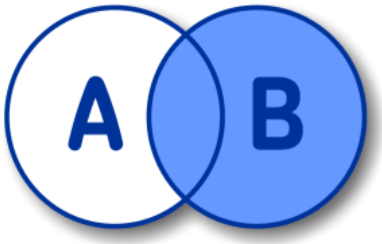
- Delete option.

- update.

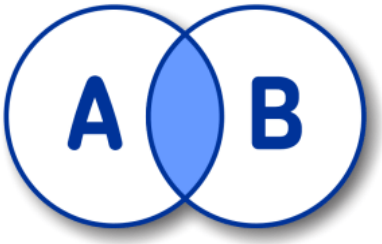
SQL JOINS



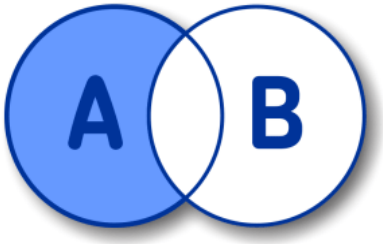
SELECT * FROM
A LEFT JOIN B
ON A.KEY = B.KEY



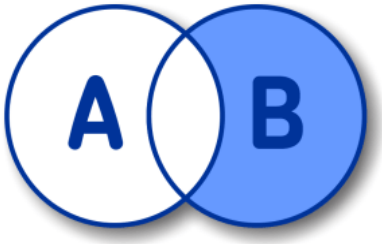
SELECT * FROM
A RIGHT JOIN B
ON A.KEY = B.KEY



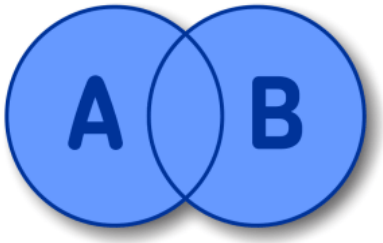
SELECT * FROM
A INNER JOIN B
ON A.KEY = B.KEY



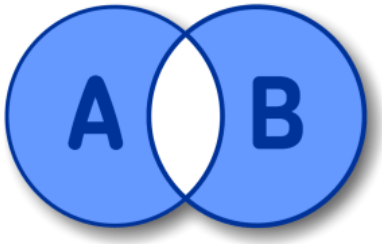
SELECT * FROM A
LEFT JOIN B
ON A.KEY = B.KEY
WHERE B.KEY IS NULL



SELECT * FROM A
RIGHT JOIN B
ON A.KEY = B.KEY
WHERE A.KEY IS NULL



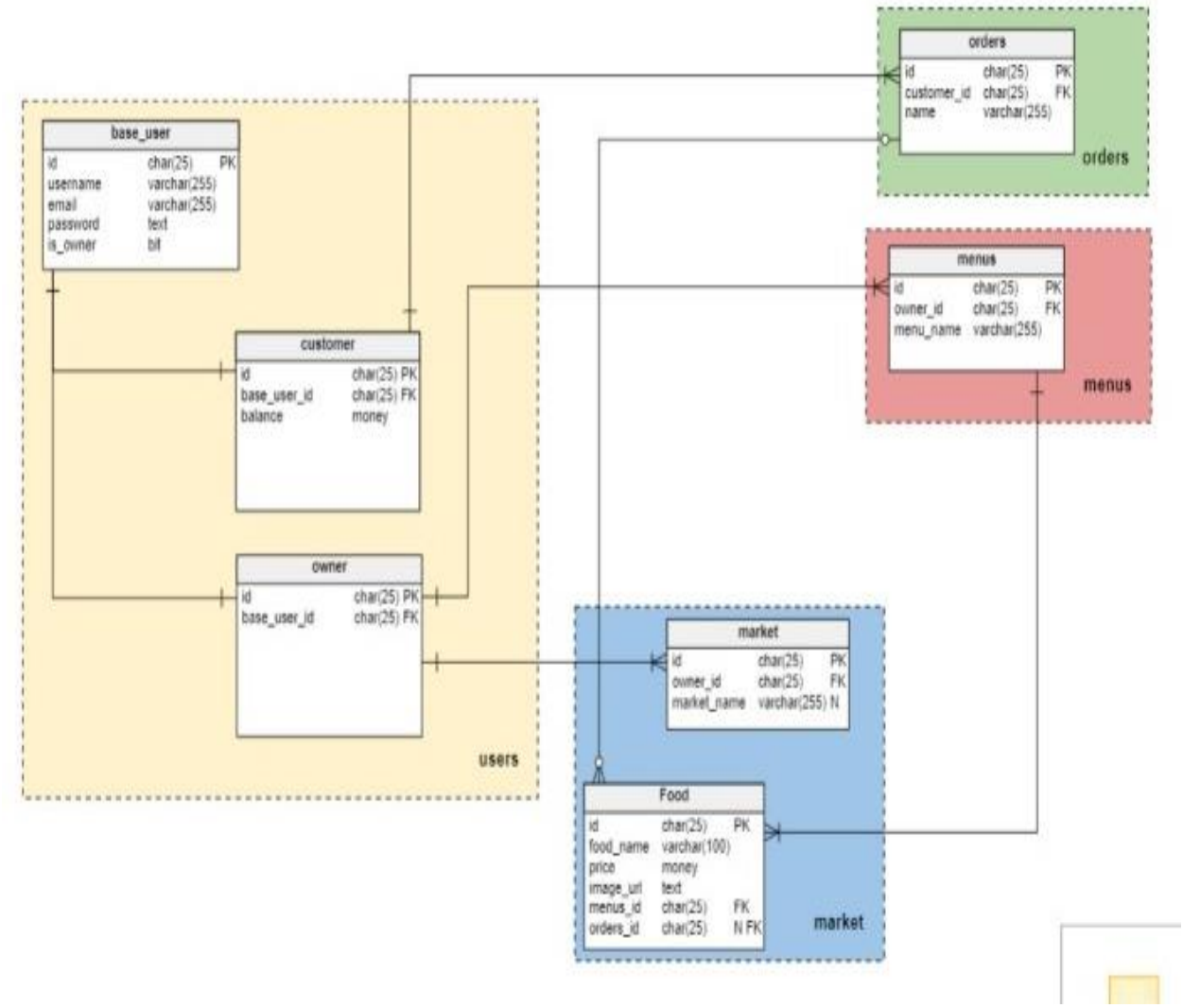
SELECT * FROM A
FULL OUTER JOIN B
ON A.KEY = B.KEY



SELECT * FROM A FULL
OUTER JOIN B ON A.KEY =
B.KEY WHERE A.KEY IS
NULL OR B.KEY IS NULL

Relational Design:

Schema:



Normalization forms:

Our System Database design guarantee that the

1NF, 2NF, 3NF are verified which can be shown through schema

1NF: Each table cell should contain a single value.

2NF: Be in 1NF & Single Column Primary Key that does not functionally dependent on any subset of candidate key relation

3NF: Be in 2NF & has no transitive functional dependencies

Transitive functional dependencies: Whenever some indirect relationship happens to cause functional dependency (FC), it is known as Transitive Dependency. Thus, if A -> B and B -> C are true, then A -> C happens to be a transitive dependency.

Thus, to **achieve 3NF**, must **eliminate** the Transitive Dependency.

Relational Schema: