

notasting

CS 340 Project Portfolio

Jacob Morrow morrowj2@oregonstate.edu

Oregon State University

Lyell Read readly@oregonstate.edu

Oregon State University

November 29, 2021

Contents

Site URL	3
Executive Summary	3
Project Design	4
Project Outline	4
Database Design	5
Database Outline	5
Inventory	5
Products	5
ProductsPurchased	5
SaleData	5
UserData	6
ThirdParties	6
Relationships	6
Entity-Relation Diagram	7
Database Schema	8
Website Screen Captures	9
Homepage	9
CREATE / READ / DELETE Inventories	10
CREATE / READ / UPDATE / DELETE Products	11
CREATE / READ / UPDATE / DELETE ProductsPurchased	12
CREATE / READ SaleData	13
CREATE / READ UserData	14
CREATE / READ ThirdParties	15

Site URL

The project can be found at: <http://128.193.54.182:4096/>

Executive Summary

Fixed issue: Removed ability to create arbitrary IDs when creating new table entries

Previously, all pages allowed the entry of the ID number, otherwise known as a primary key, within their respective INSERT forms. This was redundant as all the ID numbers were set to auto-increment. Within the most recent version, the ID numbers are now unable to be set by the user, allowing auto-increment to manage it entirely. This change was inspired by peer comments, as well as instructions from the instructor.

Fixed issue: Issues with relationships in database specification versus implementation

The original description of Notasting stated that “Notasting operates a single, central warehouse, thus there will be a single inventory.” Having a single inventory no longer matches the current direction of the project and was removed entirely from the description. This change was made in response to peer feedback that stated that it would be more realistic to have multiple locations.

Fixed issue: Nullable attribute in table, not a nullable foreign key

Originally we did not have the ability to set a nullable foreign key within any of our INSERTs, only a nullable attribute was included. Within our most recent version of Notasting, we have included an implementation for a nullable foreign key in the Product’s INSERT with the Inventory ID, which is displayed in the table below it. This was done as a result of examining peer feedback and project requirements.

Fixed issue: Did not have an intersection representing our M:M relationship

The draft for Notasting included a list to represent an attribute. The database had to be restructured entirely in order to support this addition. This created the Products Purchased table, holding the attributes Product ID and Purchase ID, to represent the many to many relationship between Products and Sales Data. This change was made in response to peer feedback and instructor feedback.

Fixed issue: Address structure in UserData

Initially, the database schema inefficiently handled user data, specifically addresses. The addresses were condensed into common components. These changes were made following instructor guidance, peer feedback and instructor feedback.

Project Design

Project Outline

Notasting, ‘totally not a CIA operation’, sells \$1 million of questionable goods a year. To support this, a database-driven website was implemented in order to record the sales data related to orders of purchased products from inventory to users whose user information is being sold to third parties.

Essentially, the database will be used to track orders and how that order information has been sold on to third party information aggregators. This is so that Notasting can understand how their operations are related to the CIA surveillance they are ‘totally not participating in’.

The database will relate Products to SaleData using ProductsPurchased, an intersection of those two entities. Further, Products will be related to Inventory in one of the central warehouses that Notasting runs. The SaleData will be associated with UserData for the user that made the purchase, and that UserData will be associated with a ThirdParty to whom that UserData was exclusively sold.

Database Design

Database Outline

Inventory

Contains details about the Products stored at a location

inventoryID	int(11)	AUTO_INCREMENT, PRIMARY KEY
location	varchar(255)	NOT NULL

Products

Contains details about Products sold in the shop

productID	int(11)	AUTO_INCREMENT, PRIMARY KEY
productName	varchar(255)	NOT NULL
description	varchar(255),	
price	decimal(8,2)	NOT NULL
inventoryID	int(11)	DEFAULT NULL, FOREIGN KEY

ProductsPurchased

Intersection between Products and SaleData representing M:M relationship

productID	int(11), FOREIGN KEY
purchaseID	int(11), FOREIGN KEY

SaleData

Contains the information associated with a purchase of Product(s)

purchaseID	int(11)	AUTO_INCREMENT, PRIMARY KEY
userID	int(11)	NOT NULL, FOREIGN KEY
total	decimal(8,2)	NOT NULL,

UserData

Contains information about a user

userID	int(11)	AUTO_INCREMENT, PRIMARY KEY
firstName	varchar(255)	NOT NULL,
lastName	varchar(255)	NOT NULL,
streetAddress	varchar(255)	NOT NULL,
zipCode	int(11)	NOT NULL,
countryCode	int(11)	NOT NULL,
thirdPartyID	int(11)	NOT NULL, FOREIGN KEY

ThirdParties

Contains the details of a third party to which UserData is sold

thirdPartyID	int(11)	AUTO_INCREMENT, PRIMARY KEY
thirdPartyName	varchar(255)	NOT NULL,
description	varchar(255),	

Relationships

M:M **Between Products and SaleData**

A single order or SaleData can contain many Products. Further, a product can be purchased several times (in different SaleData). The PKs purchaseID and productID are both FK in the intersection relation ProductsPurchased.

1:M **Between Inventory and Products**

Products can only be in a single Inventory, however an Inventory can contain multiple Products. This relationship is exemplified as the Inventory PK inventoryID being FK in Products.

1:M **Between SaleData and UserData**

A SaleData can only be purchased by one user, however a user can purchase many SaleData. This is represented by the userID (PK for User) being FK in SaleData.

1:M **Between UserData and ThirdParties**

A UserData can only be sold to one ThirdParties, however ThirdParties can buy many UserData. This is represented by thirdPartyID (PK for ThirdParties) being FK in UserData.

Entity-Relation Diagram

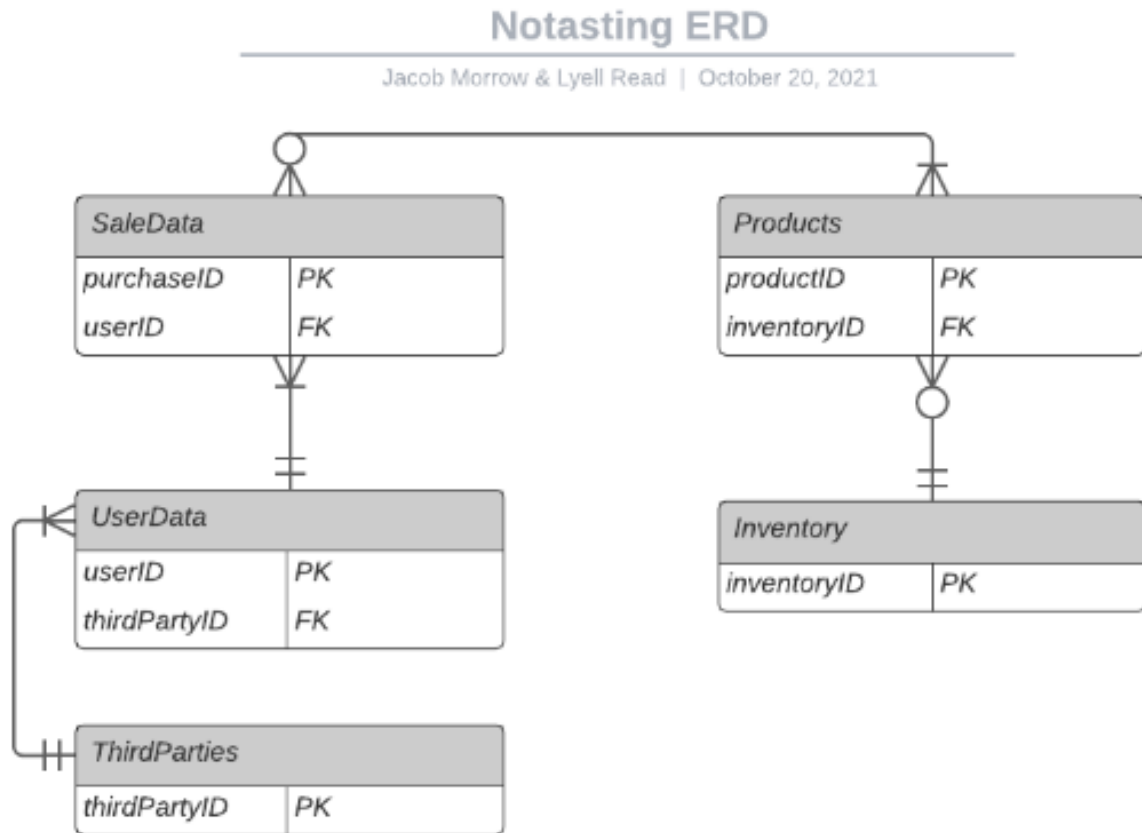


Figure 1: Entity-Relation Diagram for Notasting's Database

Database Schema

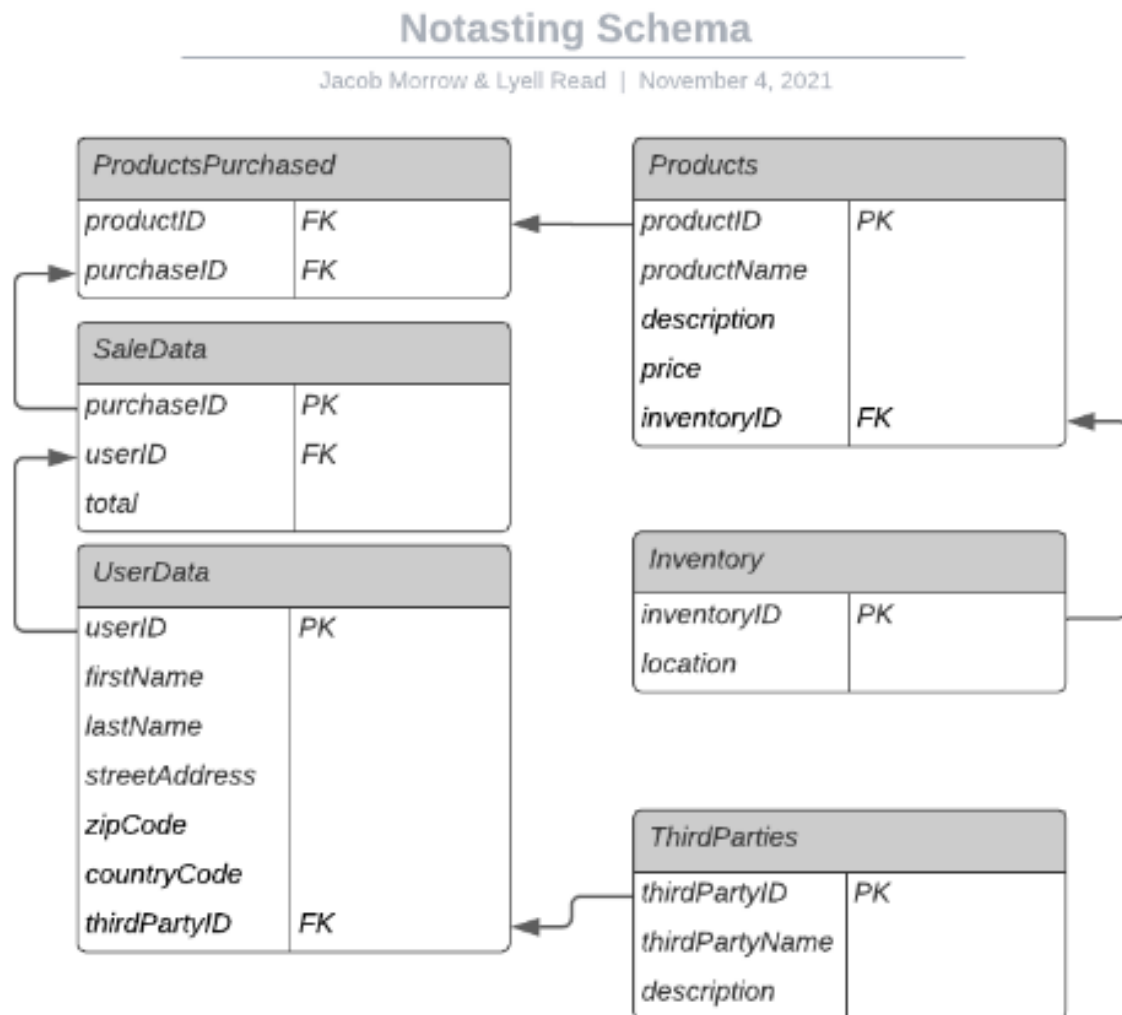


Figure 2: Database Schema Diagram for Notasting's Database

Website Screen Captures

Homepage

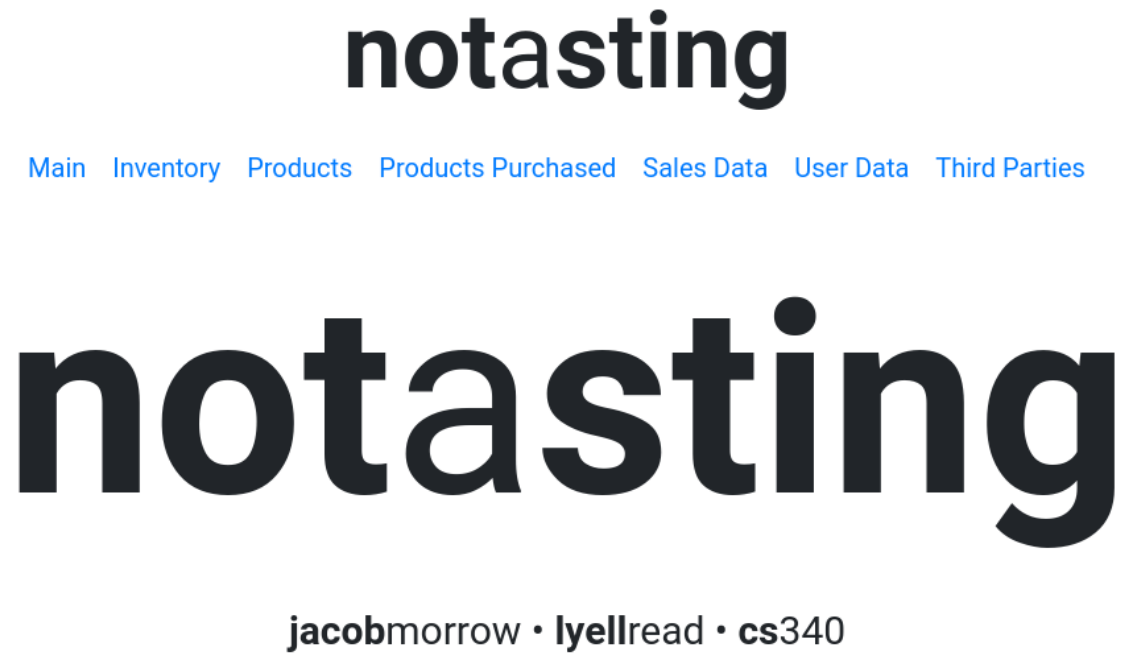


Figure 3: Home Page

CREATE / READ / DELETE Inventories

notasting

[Main](#) [Inventory](#) [Products](#) [Products Purchased](#) [Sales Data](#) [User Data](#) [Third Parties](#)

Enter a new Inventory:

Location

Enter

Inventories:

Search:

inventoryID	location	Delete Entry
1	San Miguel	Delete
2	Orlando	Delete
3	Portland, OR	Delete
4	Debanville Falls	Delete
5	Park City	Delete
6	Phoenix	Delete
7	Maine	Delete
8	Austin	Delete
9	Cape Cod	Delete
10	Spokane, WA	Delete
11	Coravillis	Delete
12	Grand Rapids	Delete
13	Arlington	Delete
15	Auburn	Delete

jacobmorrow • lyellread • cs340

Figure 4: Inventories Page

CREATE / READ / UPDATE / DELETE Products

notasting

[Main](#) [Inventory](#) [Products](#) [Products Purchased](#) [Sales Data](#) [User Data](#) [Third Parties](#)

Enter a new Product:

Product Name

Description

Price

Inventory: 1 (San Miguel) ▼

Enter

Products:

Search:

productID	productName	description	price	inventoryID	Modify	Delete
1	Bop-It	A toy that makes sounds and you have to react	4.55	1	Modify	Delete
2	Kazoo	A toy that turns your voice into a weird sound	12.50	3	Modify	Delete
3	Rubiks Cube	Puzzle Cube	5.50	3	Modify	Delete
4	Lego	A toy that makes weird sounds and you have to react	4.99	7	Modify	Delete
5	Cards Against Humanity	An offensive card game	19.99	13	Modify	Delete
6	Deck of Cards	52 fancy pieces of paper	5.98	12	Modify	Delete
7	Lego	Toy bricks	99.84	8	Modify	Delete
8	Baker's Toy Set	Learn to bake with these plastic tools	19.33	6	Modify	Delete
9	Bop-It Premium	Super Premium Bop It	9.99	10	Modify	Delete
10	Bop It	Standard Edition Bop-It	8.99	10	Modify	Delete
11	Rubik's Cube	A puzzle cube	19.99	7	Modify	Delete
12	Rubik's Cube	Puzzle Cube with 6 Sides	18.94	12	Modify	Delete
15	Kazoo Plus	A super kazoo	9.91	15	Modify	Delete

jacobmorrow • lyellread • cs340

Figure 5: Inventories Page

CREATE / READ / UPDATE / DELETE ProductsPurchased

notasting

[Main](#) [Inventory](#) [Products](#) [Products Purchased](#) [Sales Data](#) [User Data](#) [Third Parties](#)

Enter a new Product Purchased

Product ID: Purchase ID:

Enter

Products Purchased:

Search:

productID	purchaseID	Modify	Delete
1	1	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
1	2	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
2	1	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
2	2	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
10	5	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
9	2	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
7	2	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
4	4	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
8	3	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
10	8	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
6	10	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
4	11	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
2	9	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
4	12	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>
8	12	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>

jacobmorrow • lyellread • cs340

Figure 6: Inventories Page

CREATE / READ SaleData

notasting

[Main](#) [Inventory](#) [Products](#) [Products Purchased](#) [Sales Data](#) [User Data](#) [Third Parties](#)

Enter a new Sale Data

User ID:

Total

Sales Data:

Search:

purchaseID	userID	total
1	1	5.50
2	2	12.50
3	1	55.10
4	3	5.70
5	1	5.50
6	2	12.50
7	1	55.10
8	3	5.70
9	3	3.33
10	3	12.44
11	7	444.53
12	3	25.55
13	9	199.28
14	8	192.40
15	7	889.30
16	5	32.80
17	7	2.39
18	7	299.38

jacobmorrow • lyellread • cs340

Figure 7: Inventories Page

CREATE / READ UserData

notasting

[Main](#) [Inventory](#) [Products](#) [Products Purchased](#) [Sales Data](#) [User Data](#) [Third Parties](#)

Enter a new User Data Entry:

First Name

Enter firstName

Last Name

Enter lastName

Street Address

Enter streetAddress

Zip Code

Enter zipCode

Country Code

Enter countryCode

ThirdParty ID: 1 (NSA)

Enter

User Data:

Search:

Enter Search Query

userID	firstName	lastName	streetAddress	zipCode	countryCode	thirdPartyID
1	James	Apples	123 Main Street	98844	1	1
2	Dominic	Green	122 South Baker Street	22445	1	2
3	John	Doe	333 Main Street	98844	1	1
4	Jane	Doe	Park Ave Apartments, 3301 South Kane Street	98904	1	3
5	Gene	Tens	10992 SW Canary Lane, APT 02B	91002	12	2
6	Deans	Thomas	123 Main St.	98991	0	7
7	Blake	Rogers	19922 SW Beach Street, APT 199B	92111	2	8
8	Don	Thomas	10 Main ST	28394	9	5
9	Gordon	Ramsay	16 Undercooked Bass Rd.	1928	19	9

jacobmorrow • lyellread • cs340

Figure 8: Inventories Page

CREATE / READ ThirdParties

notasting

[Main](#) [Inventory](#) [Products](#) [Products Purchased](#) [Sales Data](#) [User Data](#) [Third Parties](#)

Enter new Third Party

Third Party Name

Description

Enter

Third Parties:

Search:

thirdPartyID	thirdPartyName	description
1	NSA	Not the national Security Agency, definitely not.
2	Dow Logical	Retailer of Questionable User Data
3	Facebook	This one is too easy
4	Google	...would never collect user data without your consent... right
5	Interpol	International Police
6	US DOJ	United States Department of Justice
7	ThirdPartyContractors (tm)	Definitely not NSA operating out of small businesses
8	NSA-SSD	Super Spying Division of the NSA
9	CIA	Central Intelligence Agency
10	ThirdPartyContractors (tm)	
11	Amazon Data Services	Bezos is watching too

jacobmorrow • lyellread • cs340

Figure 9: Inventories Page