

# Computer Science Task 4

Two safety measures that could be put into place for iOcean's protection of customer data are: File encryption and Authoritative Access. File encryption is the encoding of files in a way that anyone without the adequate means of decoding it, even if they have the files, will not be able to read the content. This would greatly benefit the security as even if a hacker manages to get sensitive data, they will not be able to understand or read the actual files. Authoritative Access would be another way to help protect your data. Allowing the data to be accessed only by specified users, and edited only by the system itself would lessen the chance of the data being compromised. Allowing only certain people to have access to the information, although a hacker may still find a way around that, is a lot more secure than having it open for the world to play with.

Data integrity is basically (at least in the database world) a set of rules and regulations that need to be followed to ensure the integrity of the data within the table. There are three types of integrity within databases (apart from a specific user-defined rule); Entity Integrity, Referential Integrity, and Domain Integrity. Referential integrity refers to the use of Foreign Keys in relational databases. It refers to the rule wherein every table that has a reference to another table must have the Foreign Key column responsibly for the connection as either a Primary Key from another table, or in some cases, as a null value where there is no connection. Whereas Entity Integrity refers to the Primary key of each entry, this refers to the fact that each and every record in a table must have a Primary Key assigned to it that is NOT null and is not the same as any other record within the table. Domain Integrity on the other hand refers to the fact that each column in a database must be assigned a type, or purpose. This essentially states that you cannot have a column that's just there completely empty, or only there for some (not NULL but actually not present).

It is important to have a data dictionary for our database for a couple of reasons. The first and most important one is that without it, we wouldn't have a clearly defined set of columns for our tables, we might miss data in the initial database setup or just generally not have a good idea of what data we're setting up our database to contain. Another reason would be a quick and simple reference to what is contained in the database, and also what is required for another entry. With a data dictionary it's a quick and easy process to set up forms for users to enter, to query for information, and is also a great guideline for market research. Without a data dictionary you would have to go into the database and check each column for each table etc. etc. which just wastes your time. Also in some cases, can be quite confusing.

Customer Table Data Dictionary (All the fields located in the Customers table) [CustomerID is PK]

Name	Type	Description	Length	FK	Constraints
CustomerFirstName	Alphanumeric	Customer's first name	255	N	No special characters or numbers
CustomerLastName	Alphanumeric	Customer's last name	255	N	No special characters or numbers
CustomerAddress	Alphanumeric	Customer's shipping address	65535	N	N/A
CustomerSuburb	Aplphanumeric	Customer's shipping suburb	255	N	No special characters or numbers
CustomerCity	Alphanumeric	Customer's shipping city	255	N	No special characters or numbers
CustomerMobileNum	Number	Customer's mobile phone number	9	N	First two characters = 41

Order Table Data Dictionary (All the fields located in the Orders table) [OrderID is PK]

Name	Type	Description	Length	FK	Constraints
OrderPaymentType	Numeric	Payment type enumeration number (0=PayPal, 1=Visa, 2=MasterCard, 3=Direct bank debit)	1	N	-1 < value < 4
OrderTotal	Currency	Total cost of the order	255	N	Value is positive
OrderDate	DateTime	Date the order was placed	255	N	N/A
OrderDelivered	DateTime	Date the order arrived	255	N	Value != OrderDate
CustomerID	Numeric	The ID of the customer who placed this order.	255	Y	Is valid ID from Customer table

Stock Table Data Dictionary (All the fields located in the Stock table) [StockID is PK]

Name	Type	Description	Length	FK	Constraints
StockName	Alphanumeric	Name of the stock item.	255	N	No special characters
StockBrand	Alphanumeric	Stock's brand	255	N	No special characters
StockPrice	Currency	The price of the stock item.	255	N	Value > 0

OrderedItem Table Data Dictionary (Fields located in the OrderedItem table) [OrderedItemID is PK]

Name	Type	Description	Length	FK	Constraints
OrderID	Numeric	OrderID the item belongs to.	255	Y	Is valid ID from Order table.
StockID	Numeric	ID of the stock item to send.	255	Y	Is valid ID from Stock table.

