# Task-1.1

# Database

To put it in simple terms, a database is a collection of data designed in a manner that allows managing, searching, modification and deletion of content. A database is meant to be accessed and used by a computer. Databases are used in companies, offices, schools, shops etc. these databases help make the job of managing information and records easier.

The different types of databases as stated by searchsclever.com are as follows:

“

1. [Centralized database](https://www.tutorialspoint.com/Centralised-database).
2. [Distributed database](https://www.tutorialspoint.com/Distributed-database).
3. [Personal database](https://www.tutorialspoint.com/Personal-database).
4. [End-user database](https://www.tutorialspoint.com/End-User-Database).
5. [Commercial database](https://www.tutorialspoint.com/Commercial-Database).
6. [NoSQL database](https://www.tutorialspoint.com/NoSQL-Databases).
7. [Operational database](https://www.tutorialspoint.com/Operational-Database).
8. [Relational database](https://www.tutorialspoint.com/Relational-Databases).
9. [Cloud database](https://www.tutorialspoint.com/Cloud-Databases).
10. [Object-oriented database](https://www.tutorialspoint.com/Object-Oriented-Databases).
11. [Graph database](https://www.tutorialspoint.com/Graph-Databases).”

# Database management system

To be succinct, a database is a tool design to help facilitate and help simplify the use of the database, it’s main goal is to help in creating, adding, modifying or deleting information in the database. It manages three corner stones:

1. The data: this is the information entered in the database
2. The database engine: this is what helps in accessing, viewing and modifying data
3. The database schema: this entity plays the role of defining the DB’s logical structure

These three mentioned factors together play the role of providing [concurrency](https://searchoracle.techtarget.com/definition/concurrent-processing), security, [data integrity](https://searchdatacenter.techtarget.com/definition/integrity) and uniform administration procedures

# Examples of DMBS

1. MySQL
2. SQL Server
3. Oracle
4. FoxPro
5. Dbase

# Advantages of DBMS

A DBMS’s strongest suit is acting as record providing a centralized view of data that is capable of providing multiple users regardless of their location with information at the same time, this information viewed by certain people can be limited, monitored and be viewed differently by others. A very useful trait of the DBMS is the fact that it’s capable of offering both logical as well as physical data independence meaning neither the users nor the applications need be concerned with any change with changes regarding the physical structure of the data, another advantage and the most important one is the fact that the DBMS is capable of providing access to the data while maintaining data integrity. There’s also the consistency that comes with DBMS, in a situation the data is duplicated in multiple pages and changes are made to one of the entries, the DBMS senses the inconsistency and eliminates it. Another advantage of using DBMS is the authority and security it provides in situations where multiple users access the database, certain limitations and restrictions can be placed on certain users.

(Technopedia, 2019)

# DBMS applications

DBMS’ have multiple applications such as:

1. Industry: facilities such as distribution centers, production units and warehouses require a database to monitor and keep tracks of all the transactions as well as ins and outs of the facility
2. Sales: Stores and any other retail services require databases to store information about the products, costumers, invoice descriptions and sales
3. Telecom: in the telecommunication field, a database is required to keep track of costumer details, calls and the network use
4. Online shopping: online shopping sites require databases to keep track of costumer details, addresses, costumer interests and credit card details

(beginnersbook.com, 2019)

# Task-1.2

## The 3 levels of DB Architecture

There are 3 levels of DB architecture, they are:

1. Physical level
2. Conceptual level
3. External level

## Physical level

The first level known as the physical or internal level of the DB mainly clarifies how the storage of the data takes place in the database. At the lowest level, this information is commonly stored in the form of bits on external hard drives that have a considerably large amount of folders and files in them. The physical level confers over the techniques used to encrypt and compress the data

## Conceptual level

This level resides above the physical level and contains the logical aspect of the user database while also describing the relationship between different data tables with no consideration for the data storage method within the database

## External level

This level is at the top of the hierarchy and resides atop the three and is also referred to as the viewing point. This level is tasked with displaying the needed data in the form of views only while hiding whatever is left of the data, this helps in situations where different users can view the database differently depending on their respective needs.

(beginnersbook.com, 2019)

## 

## 

# Task-1.3

## Big Data

Big data is a term used to describe a wide range of data that continues to expand at speeds that can be considered “ever-increasing”, including both the frequency and size at which the information is processed and produced as well as the nature and extent of the data points. Big data usually comes in many formats while being obtained from many multiple channels

(Investopedia, 2019)

## The 3v’s Of Big Data

The 3v’s of big data is a term used to describe the dimensions or characters that define big data, they are:

1. Volume, referring to the data quantity.
2. Variety, referring to the number of data types
3. Velocity referring to the speed of the data processing

# Applications of Big Data

In modern day and age, big data has become a major part of our lives and it plays roles that are both major and critical, such as:

1. Fast food industry: fast food has become the most common and popular choice when it comes to nutrition in the current age with millions of restaurants and franchises to choose from and these companies are all big data dependent for assistance with things such as the statistics, sales, popularity of various branches, orders and preferences of the costumers
2. Banking sector: banks tend to use clustering techniques to make important decisions. The association rule is commonly used in banks to help predict in each branch during a specific period of time. Using data science has made it incredibly easy to manage such massive amounts of information
3. Media and entertainment: big data helps in collecting information and requirements. Computer recognition and data saving for later access, can be useful in determining why certain programs were successful or unsuccessful, helps in generating new public demand analysis, artists can pick placement where they want to promote their performances

# Task-1.4

## Transaction Processing (TPS)

The term “Transaction processing” refers any transaction or processes being carried out in real time through any transaction processing system(TSP). the process begins when a user issues a request for a process to be completed, then when a query has been established from either a TPS or a related process it runs through the authorization program, the information requests and other functions crucial to the transaction.

(Techopedia.com, 2019)

Database management is personnel that has been given the task of managing the database from analyzing the needs of the company during its development phase to granting specific parts of database access to certain departments of the business, they participate in any new component installation of systems and applications and may also assist as per their role. They monitor and control the database maintaining a quality standard and security measures to ensure best practices of database standard policies. As he is in control of the database, any error encountered has be to reported and he will take proper measures to resolve the issue.

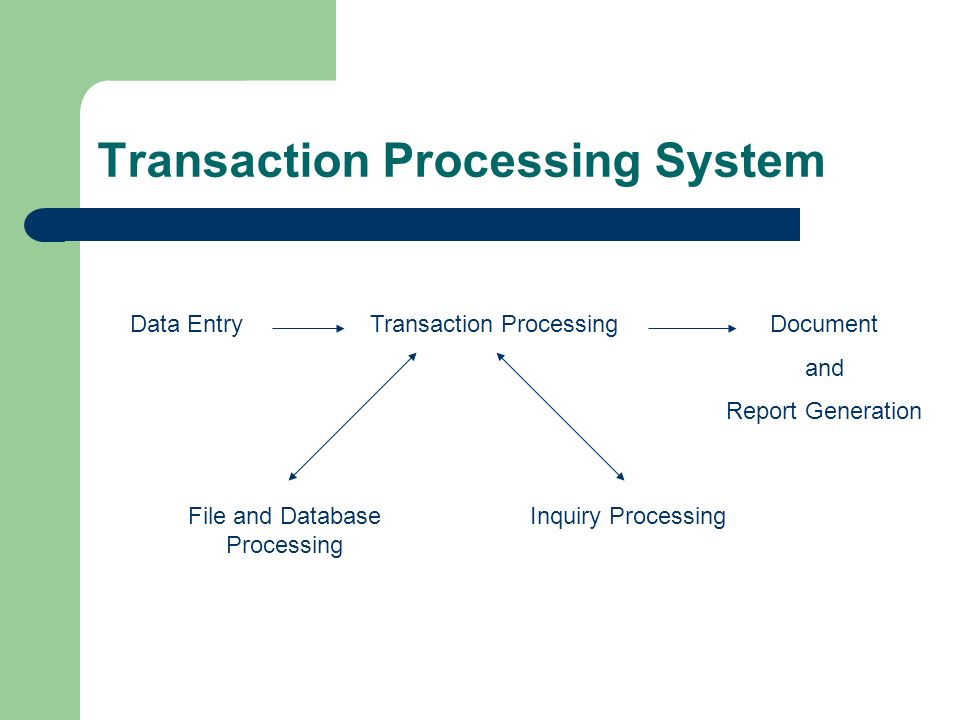


FIG-1.1: Transaction processing system (Slideplayer.com, 2019)

## Properties of transactions

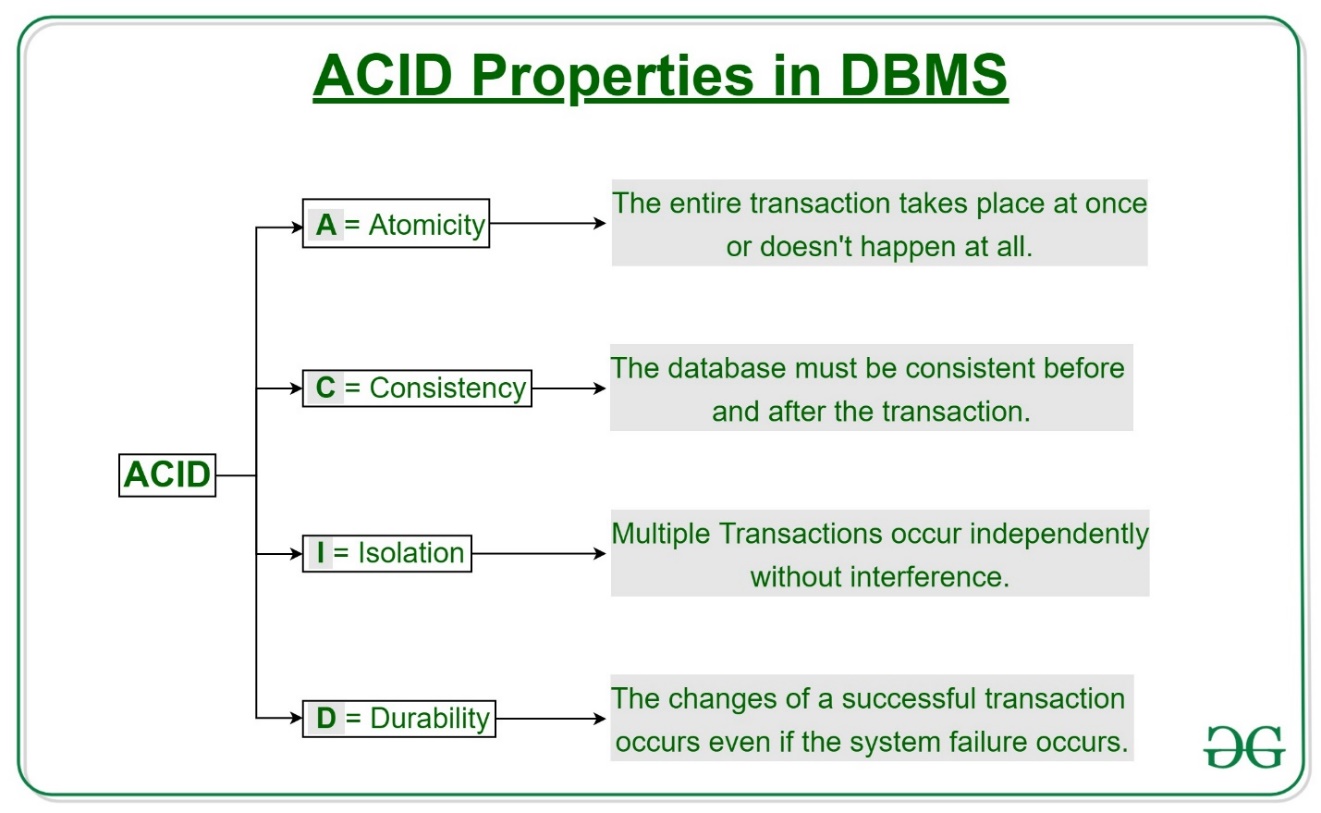


FIG-1.2: ACID properties in DBMS

ACID means that the whole cycle is either run at once or not at all. The cycle has no intermediate transactions, meaning no partial transactions. Through transaction is referred to as one unit, and is either completed or not completed. This is followed by two operations.

“Abort - Changes to a database are not visible if a transaction aborts.

Commit - Changes made are apparent when a transaction occurs.

Atomicity is also referred to as "the law of all or nothing."”



FIG-1.3: transaction

In the previous transaction, consider T, T1 and T2 transferring 100 from X to Y.

In the case that the transaction fails, following the completion of T1 but before the completion of T2 (for example, after writing(X) but before writing(Y).

It allows the server condition to be incoherent. The transaction must therefore be executed in its entirety to ensure that the server state is correct.

## **Consistency**

Integrity must be maintained in order to ensure that the database maintains it’s accuracy both before and after the transaction takes place. The database’s reliability is listed

For example, the total amount must be retained before and after the payment.

Total **before T** occurs = **500 + 200 = 700**.  
Total **after T occurs** = **400 + 300 = 700**.

The database is consistent. Inconsistency occurs when T1 is finished but T2 is not. As a result T is not complete.

# Isolation

Isolation guarantees the occurrence of multiple transactions simultaneously without the existence of any inconsistency or interference in the table. If any changes occur in any transaction, they won’t be apparent in any other transaction until these changes

# Durability

This property ensures transactions are conducted simultaneously and in some order in a condition that is equivalent to an achieved state. Those updates are in non-volatile storage now permanent. So the effects of the transaction are never lost.

In general, the ACID properties provide a basis for ensuring that a database is reliable and coherent, such that each transaction is a set

# States of Transactions

In one of the following states a transaction in a database can be

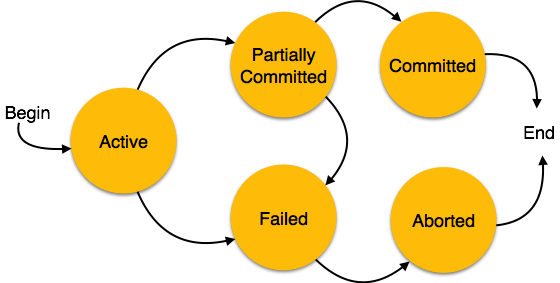


FIG-1.4- States of transaction

Active: in this state, the transaction is carried out. This is the initial condition of all transactions.

Partially Committed: a transaction is said to be a partial transaction if it carries out it’s final transaction

Failed: a transaction fails in the case that any one of the database recovery processed fail. A failed payment is no longer an option in this case

Aborted: if the transaction fails due to one of the transactions failing, all the operations are rolled back into the server by the manager to revert the database to its’s original state back before the transaction was carried out. In this case, the transaction is referred to as “Aborted”

After the abortion of the transaction, one of the following two operations can be executed by the recovery module

1. Re-start the transaction
2. Kill the transaction

Committed: when all the operations of a transaction are executed successfully, it’s referred to as committed and all the changes brought about by the transaction are constantly established

(Tutorialspoint.com, 2019)

Transaction manager: a transaction manager is a crucial part of the application tasked with the control of one or more of the resources in order to coordinate the transactions. The transaction manager usually controls every asset manager involved in the transaction.

A transaction administrator is always logged-in. Generally, the log is a serial file serving the purpose of recording the transaction events, transaction managers use this log to record transactions, commit decisions and enlistments. In an ideal scenario, the transaction managers only have to enter the log when they are handled normally, however when they fail, the log must be read instantly when it begins once again for restoration

Transaction managers have multiple responsibilities, they are as follows:

1. Demarcation: Transactions start and end with methods launch, commit and roll-back.
2. Control the context of the transaction: transaction context can be used by the transaction manager to monitor the transaction. It is the transaction manager’s responsibility to create as well as link transaction context to the current thread.
3. Co-ordination of the transaction: a transaction manager is capable of orienting the transactions over multiple resources in different orientations. This role includes the commit protocol in two phases, it can also serve the purpose of registering as well as handling assets within the XA protocol
4. Recovery from failure: if a system or an application failure occurs, transaction managers are tasked with keeping the resources from being kept in a state that can be considered inconsistent

# Task-1.5

## Data integrity

The term “Data integrity” refers to the overall accuracy, consistency and completeness of the data. This is usually visible through the absence of modifications between two instances or two updates in a data record meaning the data remains both stable and unchanged. Usually, the integrity of the data is ensured by guidelines and standard procedures during the design phase of the database, it can be established by multiple methods of validation and error checking

Data integrity is standard implementation is both hierarchal and relational database

The following integrity constraints are used in a relational database structure to achieve data integrity:

1. Entity integrity: the main concept applied here is that of the primary keys stating that each table has it’s own primary key and that the table is unique and not null
2. Referential integrity: the concept applied here is that of foreign keys stating that the value of the foreign key is capable of being in two states with the value of the foreign key is equal to that of the primary key’s or it can’t be zero. For the value to equal zero would mean that the nature of the relationship is either unknown or that there is none
3. Domain integrity: this concept revolves around the existence of a relationship between each column in a database. The concept of data integrity ensures the traceability and connection of all data within the database meaning that all the data is searchable and recoverable. The reliability, efficiency, reusability and maintenance can be improved within a single managed and well specified framework for data integrity. If the database is incapable of having any one of these features implemented, it has to be done through software

## Importance of data integrity

It’s a known fact that data is a factor that can’t be considered constant, meaning it can be altered multiple times throughout it’s lifetime by going through processes such as generation, change and deletion whether it be done intentionally or unintentionally. Data also acts as the driving force behind any and all devices, a device without any data can be considered useless. The components of a device are capable of predicting the various inputs and producing the corresponding output. The term known as “device failure” occurs when the data that the device receives doesn’t match what it should have received. Data integrity plays the role of ensuring the smooth processing in the system

## GIGO

The term GIGO which is an acronym that stands for “Garbage in, garbage out” which is used to describe a failure in the system resultant of a system failing to interpret bad data obtained which leads to a result that’s deemed incorrect and in some conditions may lead to a crash. An example would be a program trying to access unsuitable data and the kernel not allowing it to do so resulting in a crash or an abnormal termination.

In simple terms, GIGO describes a situation where poor performance is seen due to the intake of poor data.

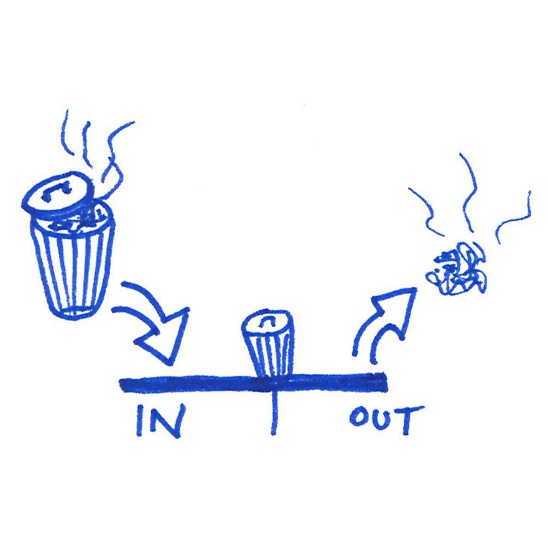


FIG-1.5: GIGO representation

## Data quality

The ability to fulfil an intended purpose of a given set of data is known as data quality, in other terms if the output provided by the data is as expected, that is referred to as data quality. If you lack data quality however, you will be unable to use the data as you please

Here are a few real world examples of data quality challenges to help illustrate the term

In a scenario where we have a set of data consisting of addresses and names, there are bound to be certain basic and complicated mistakes. These errors might occur due to the data being entered incorrectly or having it change since being collected.

## Fixing data quality problems

The most basic and simple way to correct these problems would be to manually inspect and patch any and all confusion or inconsistencies, however that would be very time consuming and in some cases can be considered impossible, therefore it would be best to use an automated data quality system capable of data detection ,analyzation and fixing without human involvement in the process. In the previously mentioned scenario, this can be achieved through correlating the data with the other data sets in order to detect anomalies. It can also be done by the use of predictive analytics to help fill in the blanks.

## The never-ending data quality battle

The exact nature and characteristics of data quality that differ from case to case, due to the definition of data quality in terms of the capacity of a data set to fulfill a given mission. Within another corporation, garbage may be what one organization perceives as high quality data.

Understanding how the changes in data quality based on context are relevant because it means data quality is not just something you can get and keep.

Think of data quality as a never-ending fight. It's something you need to continually work on and improve to make sure that your data is ready to tackle any tasks that you throw at it.

# Task-2

## 2.a) ERD and Data Dictionary

### Entity Relationship Diagram-ERD

**Identifying Entities and their attributes**

6 entities make up the Database

Customer

It has 4 fields with Cust\_ID being the primary key

Cust\_ID – ID of the customer identifying the customer.

Cust\_Name – Name of the Customer.

D\_O\_B – Date of Birth of the customer.

Gender – Gender of the customer

ZIP – ZIP Code or Postal code for the customer.

Employees

It has 5 fields with Emp\_ID being the primary key

Emp\_ID – ID of the Employee identifying the Employee.

Emp\_Name – Name of the employee.

Emp\_Contact – Contact of the employee.

Emp\_Address – Address of the employee.

Suppliers

It has 4 fields with Sup\_ID being the primary key

Sup\_ID – ID of the Supplier identifying the supplier.

Sup\_Name – Name of the supplier.

Sup\_Address – Address of the supplier.

Sup\_contact – Contact of the supplier.

Raw Materials

It has 3 fields with RM\_ID being the primary key

RM\_ID – ID of the raw material identifying the raw material.

RM\_Name – Name of the raw material.

RM\_UP – Unit Price of the raw material.

Products

Prod\_ID – ID of the product identifying the product.

Prod\_Name – Name of the Product.

Prod\_Price – Price of the product.

Orders

Order\_ID – ID of the order identifying the product.

Prod\_ID – ID of the product ordered

Order\_quantity – Quantity of the products placed in the order.

**Identify the relationships**

Customer places orders

Order includes products.

Employees make Products.

Products use raw materials.

Suppliers provide Raw materials.

**Mapping**

One Customer can place many orders.

One order can include many products.

One Product can use many raw materials.

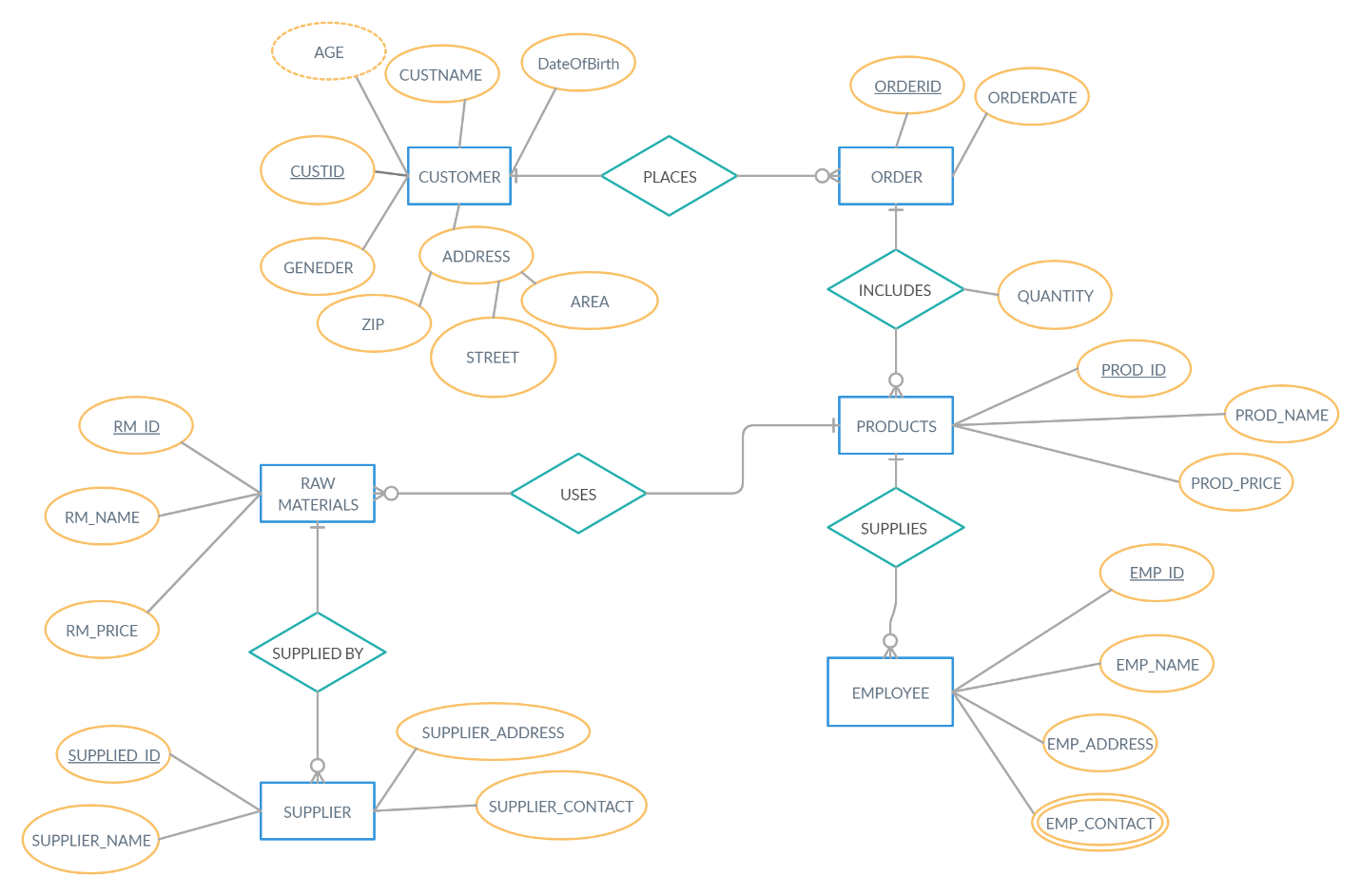
One raw material can be used for many products.

One supplier supplies one product.

One Employee makes only 1 product

One product is made by only 1 employee.

**Entity Relationship Diagram for the scenario**



### Data Dictionary

Data dictionary is called the collection of metadata descriptions of the different entities, objects and attributes in a data model. This means, this dictionary is data for the benefit of the programmers and others who need to refer to it.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 1 Customer** | | | | |
| **Field** | **Optional** | **Format** | **Length Limit** | **Description** |
| Cust\_ID | N | NUMBER | - | Primary key |
| Cust\_Name | N | TEXT | 50 | **-** |
| D\_O\_B | Y | DATE | - | - |
| Gender | Y | M/F | - | - |
| ZIP | N | NUMBER | - | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 2 Employees** | | | | |
| **Field** | **Optional** | **Format** | **Length Limit** | **Description** |
| Emp\_ID | N | AUTO NUMBER | - | Primary key |
| Emp\_Name | N | TEXT | 50 | - |
| Emp\_Address | N | TEXT | 50 | **-** |
| Emp\_Contact | N | NUMBER | - | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 3 Suppliers** | | | | |
| **Field** | **Optional** | **Format** | **Length Limit** | **Description** |
| Supplier\_ID | N | AUTO NUMBER | - | Primary key |
| Supplier\_Name | N | TEXT | 50 | - |
| Supplier\_Address | N | TEXT | 50 | **-** |
| Supplier\_Contact | N | NUMBER | - | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 4 Raw Materials** | | | | |
| **Field** | **Optional** | **Format** | **Length Limit** | **Description** |
| RM\_ID | N | AUTO NUMBER | - | Primary key |
| RM\_Name | N | TEXT | 50 | Foreign key |
| RM\_UP | N | NUMBER | - | - |

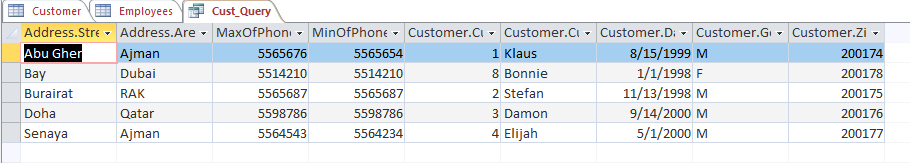
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 5 Products** | | | | |
| **Field** | **Optional** | **Format** | **Length Limit** | **Description** |
| Prod\_ID | N | AUTO NUMBER | - | Primary key |
| Prod\_Name | N | TEXT | 50 | Foreign key |
| Prod\_Price | N | NUMBER | - | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 6 Orders** | | | | |
| **Field** | **Optional** | **Format** | **Length Limit** | **Description** |
| Order\_ID | N | AUTO NUMBER |  | Primary key |
| Order\_Date | N | DATE |  | **-** |
| Cust\_ID | N | NUMBER | - | - |

## 2.b) Normalization

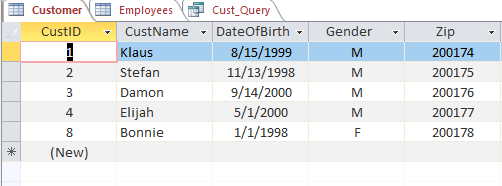
### 1st Normal Form

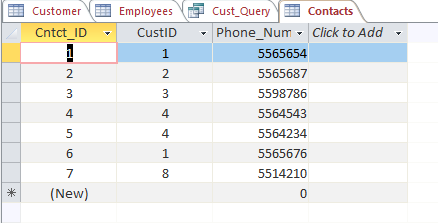
Consider Customer Table,



Customer table was not in 1NF as it had multivalued attribute- Cust\_Contact ,

so it decomposes into Customer table and Customer\_Contacts table.

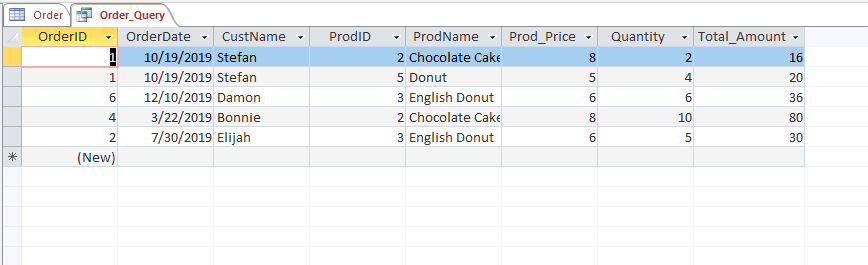




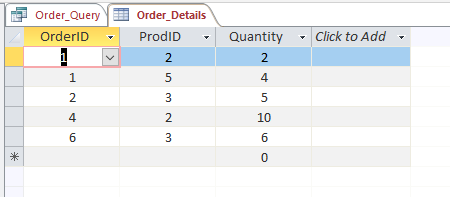
Now both tables are in 1NF.

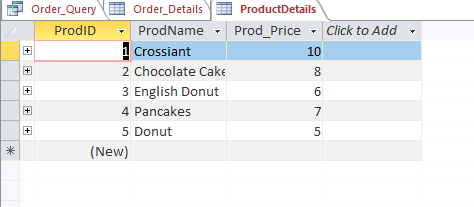
### 2nd Normal Form

Consider Order Table



Order table was not in 2NF as there is a partial dependency of Prod\_Name and Prod\_Price on Prod\_ID, so it decomposes into Order\_Item table and Products table

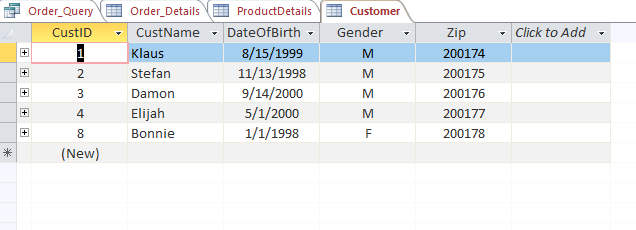
****



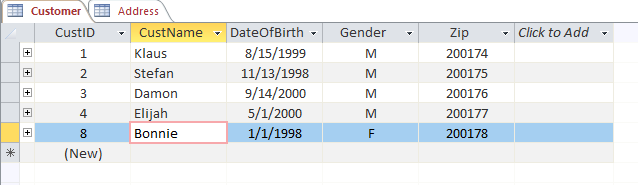
Now both tables are in 2NF

### 3rd Normal Form

Consider Customer table



Customer table was not in 3NF as there is a transitive dependency of Street and Area on Zip, so it decomposes into Customer table and Cust\_Address table.



## 2.c) Security and Data Integrity Features

### Data integrity

**Entity Integrity**

**Null Rule**

* If you did not enter anything in Zip\_Code, Street & Area in Cust\_Address table you’ll get a null value and an error will appear.
* If you did not enter anything in Cntct\_ID & Cust\_ID in Cust\_Contacts table you’ll get a null value and an error will appear.
* If you did not enter anything in Cust\_ID, Cust\_Name & Zip in Customer table you’ll get a null value and an error will appear.
* If you did not enter anything in Emp\_ID, Emp\_Name, Emp\_Address & Emp\_Contact in Employees table you’ll get a null value and an error will appear.
* If you did not enter anything in Order\_ID, Order\_Date & Cust\_ID in Order\_Details table you’ll get a null value and an error will appear.
* If you did not enter anything in Order\_ID, Prod\_ID & Quantity in Order\_Item table you’ll get a null value and an error will appear.
* If you did not enter anything in Prod\_ID, Prod\_Name & Prod\_Price in Products table you’ll get a null value and an error will appear.
* If you did not enter anything in RM\_ID, RM\_Name, RM\_UP in Raw\_materials table you’ll get a null value and an error will appear.
* If you did not enter anything in Sup\_ID, Sup\_Name & Sup\_Contact in Suppliers table you’ll get a null value and an error will appear.

**Unique Column**

Unique columns are ones which have no duplicate values.

* Zip\_Code column & Street column in Cust\_Address table are unique columns.
* Cntct\_ID column & Cust\_number column in Cust\_Contacts table are unique columns.
* Cust\_ID column in Customer table is unique column.
* Emp\_ID column & Emp\_Contact column in Employees table are unique columns.
* Order\_ID column in Order\_Details table is unique column.
* Order\_ID column in Order\_Item table is unique column.
* Prod\_ID column & Prod\_Name column in Products table are unique columns.
* RM\_ID column & RM\_Name column in Raw\_materials table are unique columns.
* Sup\_ID column, Sup\_Name column & Sup\_Contact column in Suppliers table are unique columns.

**Primary Key**

* The primary key in the Cust\_Address table is Cust\_Address
* The primary key in the Cust\_Contacts table is Cust\_Contacts
* The primary key in the Customer table is Customer
* The primary key in the Employees table is Employees
* The primary key in the Order\_Details table is Order\_Details
* The primary key in the Order\_Item table is Order\_Item
* The primary key in the Products table is Prod\_ID
* The primary key in the Raw\_materials table is RM\_ID
* The primary key in the Suppliers table is Sup\_ID

**Domain integrity**

**Data Type**

* Data type of Zip\_Code is Number, Street is short text & Area is short text in Cust\_Address table.
* Data type of Cntct\_ID is Auto Number, Cust\_ID is number & Cust\_Number is Number in Cust\_Contacts table.
* Data type of Emp\_ID is Auto Number, Emp\_Name is short text, Emp\_Address is long text & Emp\_Contact is long number Employees table.
* Data type of Cust\_ID Auto number, Cust\_Name is short text, D\_O\_B is Date/Time, Age is number, Gender is short text & Zip is number in Customer table.
* Data type of Order\_ID Auto number, Order\_Date is Date/Time & Cust\_ID is number in Order\_Details table.
* Data type of Order\_ID is number, Prod\_ID is number & Quantity is numberin Order\_Item table.
* Data type of Prod\_ID is Auto number, Prod\_Name is short text, Prod\_Price is number, RM1 is number, RM2 is number by\_Emp\_ID is number in Products table.
* Data type of RM\_ID is Auto number, RM\_Name is short text, RM\_UP is number & Sup\_ID is number in Raw\_materials table.
* Data type of Sup\_ID is Auto number, Sup\_Name is short text Sup\_Address is short text & Sup\_Contact is number in Suppliers table.

**Referential Integrity Rules**

* Customers\_ID (primary key) in Customers table is the foreign key in the Cust\_Address table, Cust\_Contacts table & Order\_Details table.
* Zip\_Code (primary key) in Customer\_Address table is the foreign key in the Customer table.
* Emp\_ID (primary key) in Employees table is the foreign key in the Products table.
* Prod\_ID (primary key) in Products table is the foreign key in the Order\_Item table.
* Sup\_ID (primary key) in Suppliers table is the foreign key in the Raw\_materials table.

### Data Dictionary

Data dictionary is the collection of metadata descriptions of the different entities, objects and attributes in a data model. This means, this dictionary is data for the benefit of the programmers and others who need to refer to it.

**Database Security & DBA Roles**

The term database security refers to the accumulative effort and the methods used to secure and protect the software from illegitimate use and malicious threats

It’s a general term to describe a group of systems methodologies and methods used to guarantee safety and security in a database setting

Database security covers and strengthens security in every other way. This includes

* Data stored in a database
* Database server
* Database management system (DBMS)
* Other database workflow applications

Database security measures tend to be planned, applied and also maintained by someone known as a database administrator or by the other information security expert

Some of the database security methods include:

1. Not allowing unauthorized access by implementing the use of highly capable restriction control
2. Securing the software from thefts and natural disasters

# Task-3

## Entity relationship diagram

### Entity relationship model

The term Entity Relationship Diagram (ERD) refers to a conceptual view of a database. It consists of associations and entities and is commonly used to design and debug databases. ERD contains various symbols such as diamond, flow lines, rectangles etc. ERDs are typically used to design and model databases and are based on the entities along with their attributes and relationship between entities

Entity

Entity

Relationship

FIG-1.6: ERD representation

## Entity

An entity is a term used to refer to a place, a name or any other noun, the properties of these entities are called attributes, attributes are also defined as a set of values.

Entity keys

The term entity key is used to refer to entities that are uniquely identified by certain attributes within the ERD. Examples of entity keys are foreign keys and candidate keys

## Entity sets

Entity sets are a set of entities of the same type sharing identical properties. It can be stored either within the table or even in the form of a relation

## Attributes

The properties of an entity are referred to as an attribute and are usually represented by either an oval or an ellipse. Each ellipse within the ERD represents a property of entity and is directly connected to it through a line

## Types of attributes

* Simple attribute: A simple attribute is one that only contains a single part and is represented by an ellipse
* Composite attribute: A composite attribute is an attribute containing multiple sub attributes
* Multi valued attribute: As the name suggests, a Multi-valued attribute is an attribute containing multiple values
* Derived attribute: A derived attribute is an attribute that can be calculated and who’s value may be affected with time

## Relationship

A relationship is defined within the ERD as a logical association in between the various entities and is represented through diamond symbol connected to the attributes through a line

Writes

Student

Exam

FIG-1.7: Relationship

## Degree of a relationship

Degree of a relationship is a term used to describe the number of entities involved in the relationship. There are mainly two types of degrees of a relationship:

1. Unary Relationship: A unary relationship is one that is maintained within the ERD, it may also be referred to as a recursive relationship. In this kind of relationship, the relationship is maintained by one entity, that entity being the machine operator.
2. Binary relationship: In this type of relationship, two entities are either connected or related to each other through a relation. This association may be from one entity to another or from one entity to many
   * 1. One-to-one: if more than a single instance entity is connected through a relationship, it is labeled as 1:1 insinuating that only a single entity is to be associated within the relationship, thereby depicting a one-to-one relationship
     2. One-to-many: if multiple instances entities are involved in the relationship, it is marked as 1:M depicting the existence of a relationship between one instance of the first entity to multiple instances if the second entity

## Relationship diagram within the Database

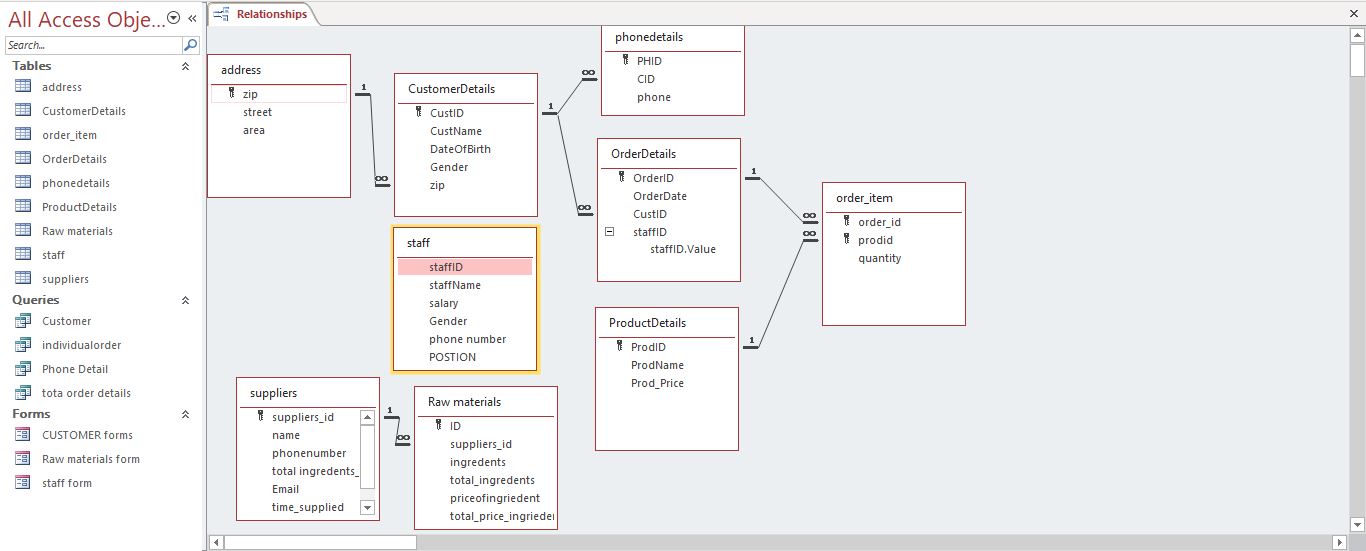


FIG-1.8: Relationships within the database

# Task-3.b

## Normalization and functional dependency

In order to avoid inconveniences such as conflict and repetition, databases are often normalized which is a method of database designing and it includes five main variants. Normalization is done with two specific goals in mind, they are

1. Removal of unnecessary data
2. Facilitation of an easy retrieval of previously deleted data

Normalization is a process that makes the database maintenance and easy task. Typically, database normalization is a process implemented after the database’s logical design while the conceptual design is being created. The normalization process is one that consists of different phases, they are:

1. Functional data dependency:

An important part of normalization is functional data dependency which is defined as a form of a relation among the various attributes within the table. If C and D are the attributes or set of the attribute L we say that D is functionally depended on C If each value of C in L has related value. We write functional dependency as

C D AND this can be read as C is functionally determine B

In functional dependency the left side is called determinants and the right side is known dependents

For example: employee there is a relation between the attribute. Functional dependency can be established like

Employee (employee Id, name, age, DOB)

Employee id name

Employee id age

Employee id DOB

1. Transitive dependency

CONSIDER attributes M, N, and O

Another attribute is carried across through transitive dependency. The focus is on relation candidate key and removing the transitive dependency when one non key characteristic defines another non key in third normal form. M, N, and O are the qualities mentioned above.

We CAN SAY THAT O is transitively dependent on M through N.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Staff\_Id** | **Staff name** | **Department num** | **Department name** |  |

Department num is transitively dependent on staff id via department number.

Partially dependency (b1,b2) b2 d

If A is the chunk of the candidate key attribute c depends on attribute d then it is known as partial dependency.

## First normal form

If there are no recurring groups in a table, it is considered to be in first normal. It means that each row should only have one value for each characteristic. Multivalued attributes are not permitted in the first normal form. The attribute should be atomic, which means it can't be subdivided any more. To achieve the initial normal form, we divide the data into its associated groups. The user must ensure that the data is not loose in order to obtain the first normal form

**1st Normal form**

**Customer Table**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CustID** | **CustName** | **DateOfBirth** | **zip** | **street** | **Area** | **Phone1** | **Phone 2** |

Rules: break the table in to two and relay them with each other

Delete the repetitive data.

The other forms are done in the same way removing the repetition.

The table is broken down in to two table’s customer table and customer phone details

**Customer**

Cust\_ID -> CustName, DateOfBirth, gender, Zip, Street, Area

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CustID** | **CustName** | **DateOfBirth** | **Zip** | **street** | **Area** |

**Customer\_phone number**

CustID -> Phone

|  |  |
| --- | --- |
| **CustID** | **Phone** |

## Second normal form

All partial dependencies must be deleted in the second Normal form. To remove partial dependency, there must be a composite key and just one component of the key is required to identify one or more attributes. By building two narratives from one, the user should project out into its own table.

**2nd Normal form**

**Individual order table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **orderiD** | **Prod id** | **prodName** | **Prod\_price** | **Quantity** | **Amount** | **Staff id** |

orderD, prodiD ->prod NAME, Prod\_price Quantity, Amount.

Prod Item is not 2NF since there is a partial dependency of prod name and prod\_Price on prodID.

So

product item

|  |  |  |
| --- | --- | --- |
| **OrderID** | **prodID** | **Quantity** |

orderiD, prodID ->Qty.

**prod**

|  |  |  |
| --- | --- | --- |
| **prodID** | **prodName** | **Amount** |

prodID -> prodName, amount.

## Third normal form

The transitive functional dependencies are removed in the third normal form. Transitive functional dependency happens when one non-key attribute defines another non-key attribute, similar to partial functional dependency. The transitive functional dependency is eliminated by moving the transitive dependent attributes to a different table while keeping the determinate as the table's foreign key.

**3rd Normal Form**

**Customer table**

CustID->CustName,DateOfBirth ,gender ,Zip, Street, Area.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CustID** | **CustName** | **DateOfBirth** | **gender** | **Zip** | **Street** | **Area** |

It is not in 3rd Normal Form because the table has **transitive dependency**

CustID -> Zip & Zip -> street and area so it decomposes into Customer table and Address table.

The above table is broken down intto Customer table and Address table.

**Customer**

CustID->Custname, DateOfBirth, Zip

|  |  |  |  |
| --- | --- | --- | --- |
| **CustID** | **CustName** | **DateOfBirth** | **Zip** |

**Address**

**Address table**

Zip -> Area, Street

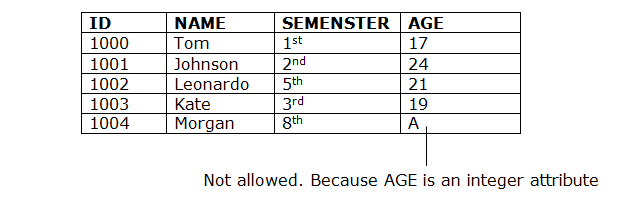
|  |  |  |
| --- | --- | --- |
| **Zip** | **Area** | **Street** |

# Task-3.c

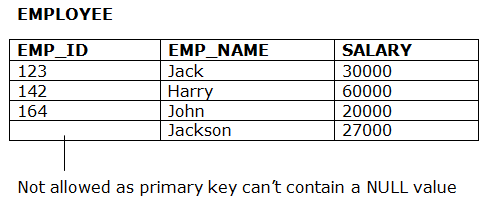
## Integrity constraints

Integrity constraints are the set of rules laid down to ensure the quality and integrity of the data. The different types of integrity constraints are as follows:

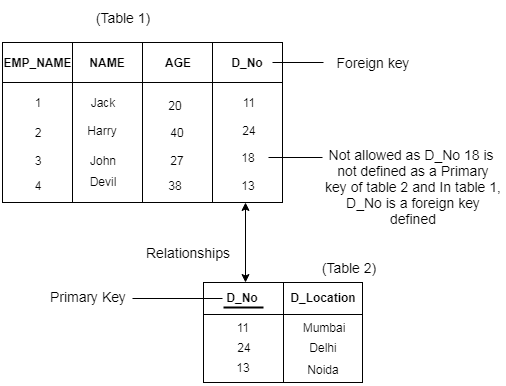
1. Domain constrains: domain constraints are basically a group of values or rules belonging to an attribute and is usually portrayed in the form of currency, time and so on.



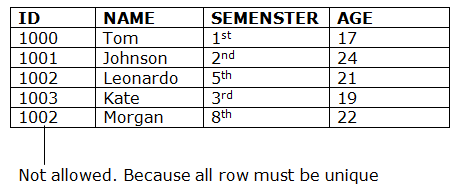
1. Entity integrity constraints: entity integrity constraints are the set of rules stating that a primary key’s value can’t be empty or equal to zero



1. Referential integrity constraints: referential integrity constraints are the set of rules that are given specifically to the relationships between a group of tables, Meaning if the table 1 contains reference to primary key of table, then all data related to the second table existing in the table 1 must be zero or empty



1. Key constraints: These rules state that every entity must have many keys, including the primary key, in order to be identified on a unique basis.



# Task-3.d

## Data integrity and security control:

Data integrity refers to the preservation of data and information by guaranteeing that it is stored and secured in a consistent and accurate manner. The overall stored data is secured by assuring the data integrity's security. Security control is the process of ensuring data security in this way. The following are the numerous types of data integrity:

1. Physical integrity: Physical integrity refers to ensuring data security and protecting it from natural calamities as well as human faults.
2. Logical integrity: Logical integrity safeguards data against being hacked or any form of error occurring during its use.

There are mainly three types of security controls used to protect the data, they are:

1. Physical control: Physical control is a type of security control in which data is safeguarded by hardware devices such as security cameras, antivirus software, and firewalls, among other things.
2. Technical control: Technical control entails the use of hardware or software protection mechanisms to keep data safe from errors or viruses.
3. Administrative control: Administrative controls are a type of security control that use a set of rules to safeguard data against errors.

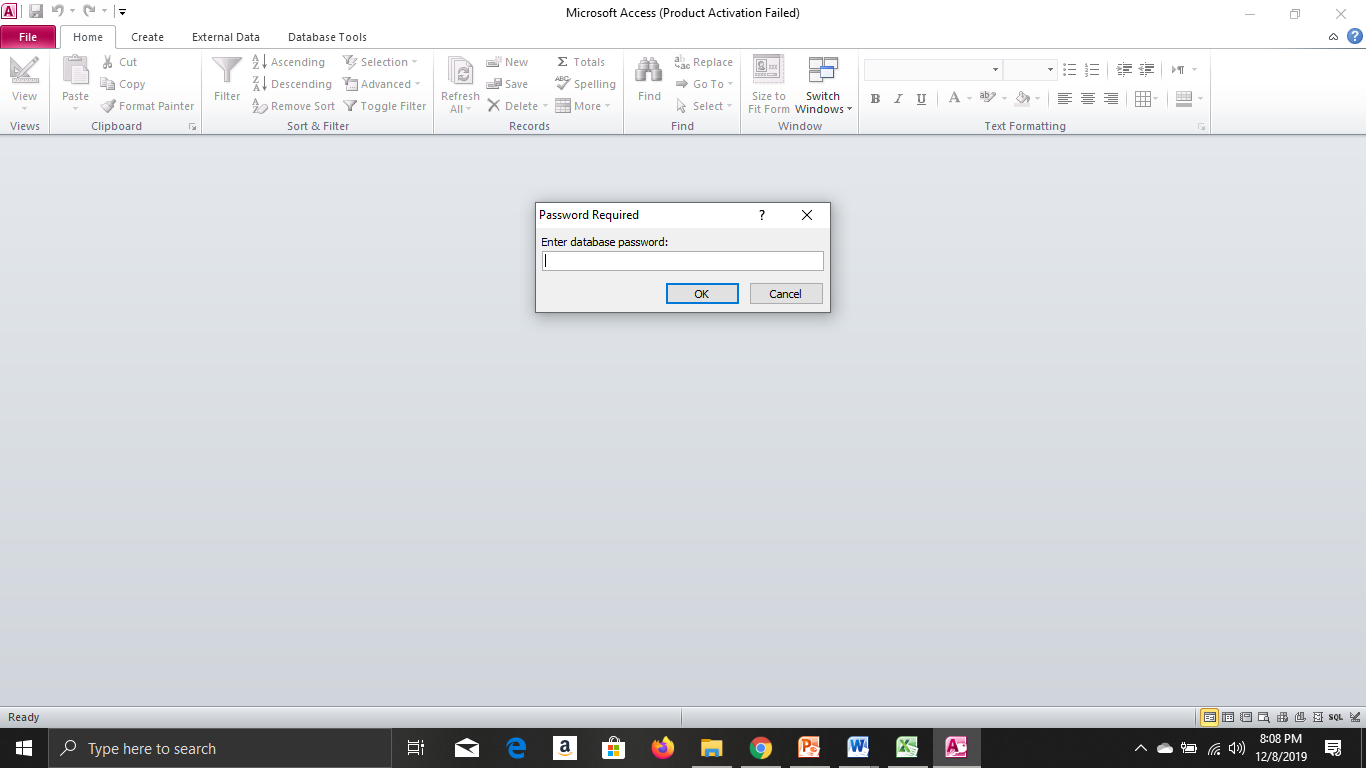


FIG-1.8: Administrative control

# Task-3.e

In this task, the ways in which the program meets the client brief will be displayed

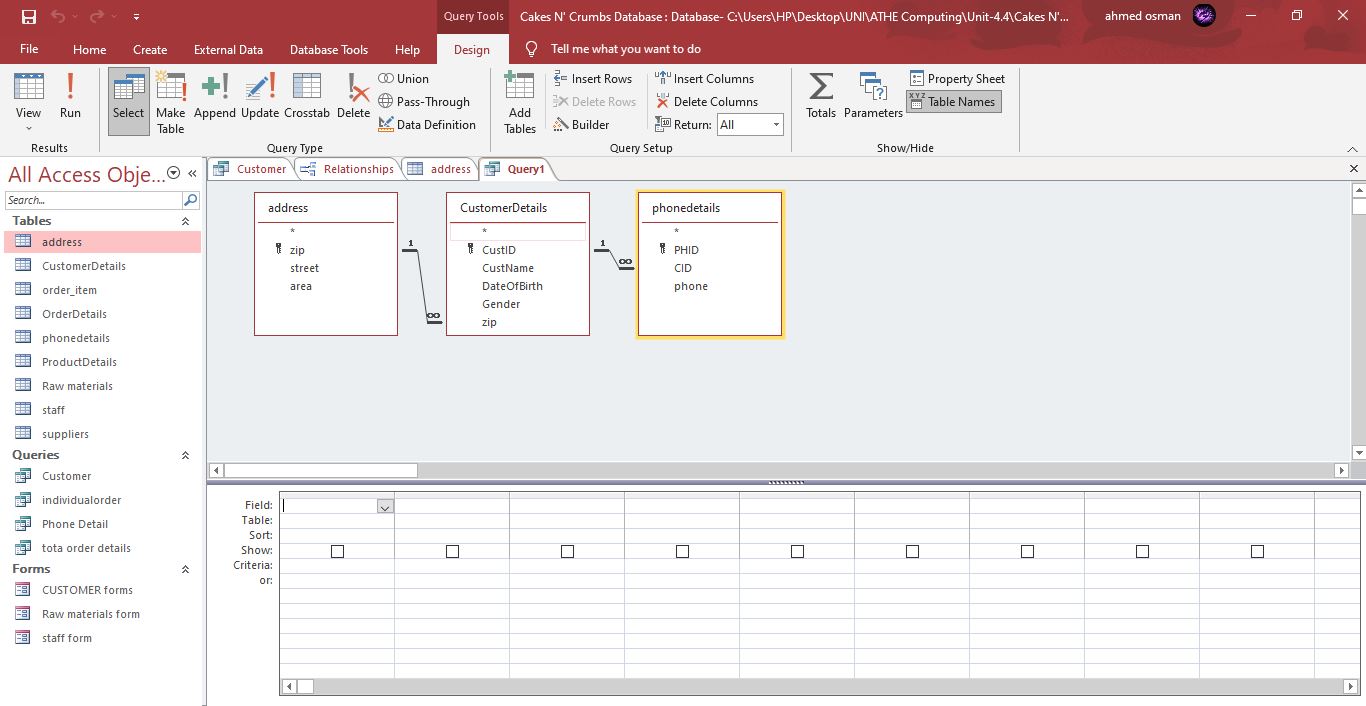


FIG-1.9: Query 1-Customer

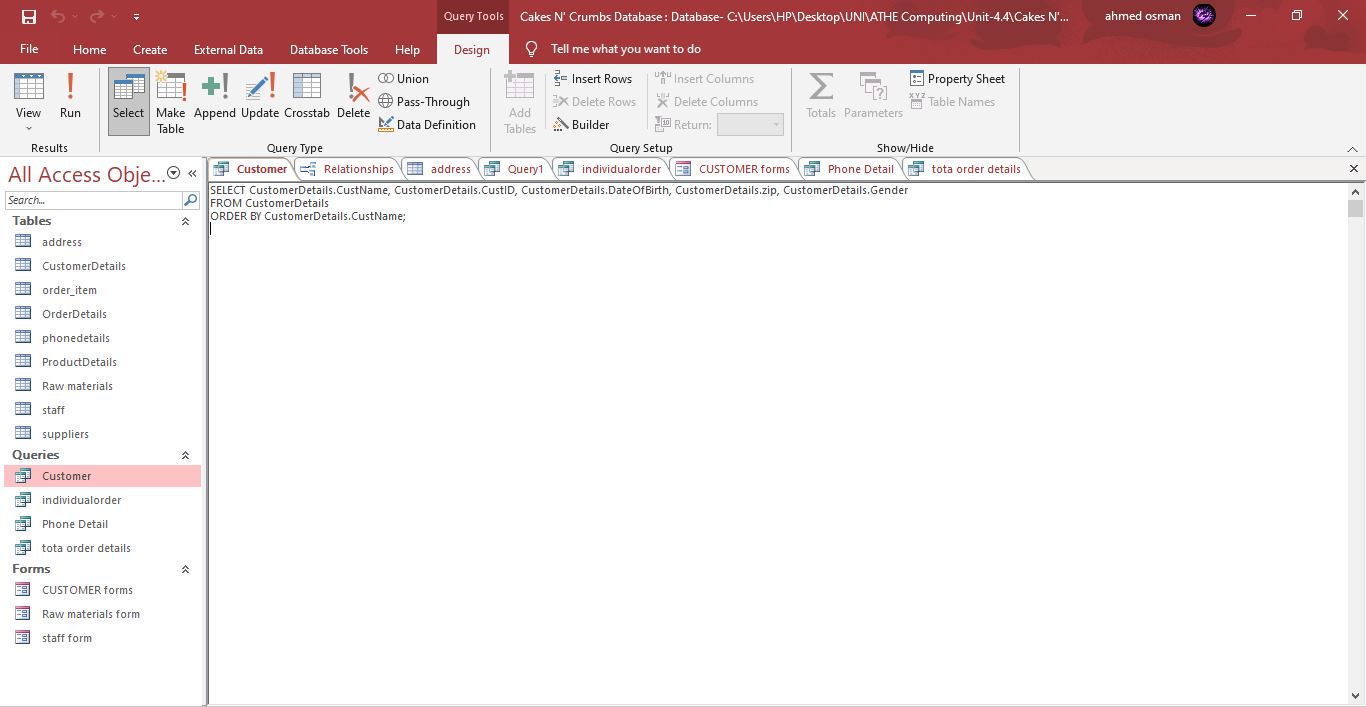


FIG-1.9: SQL View

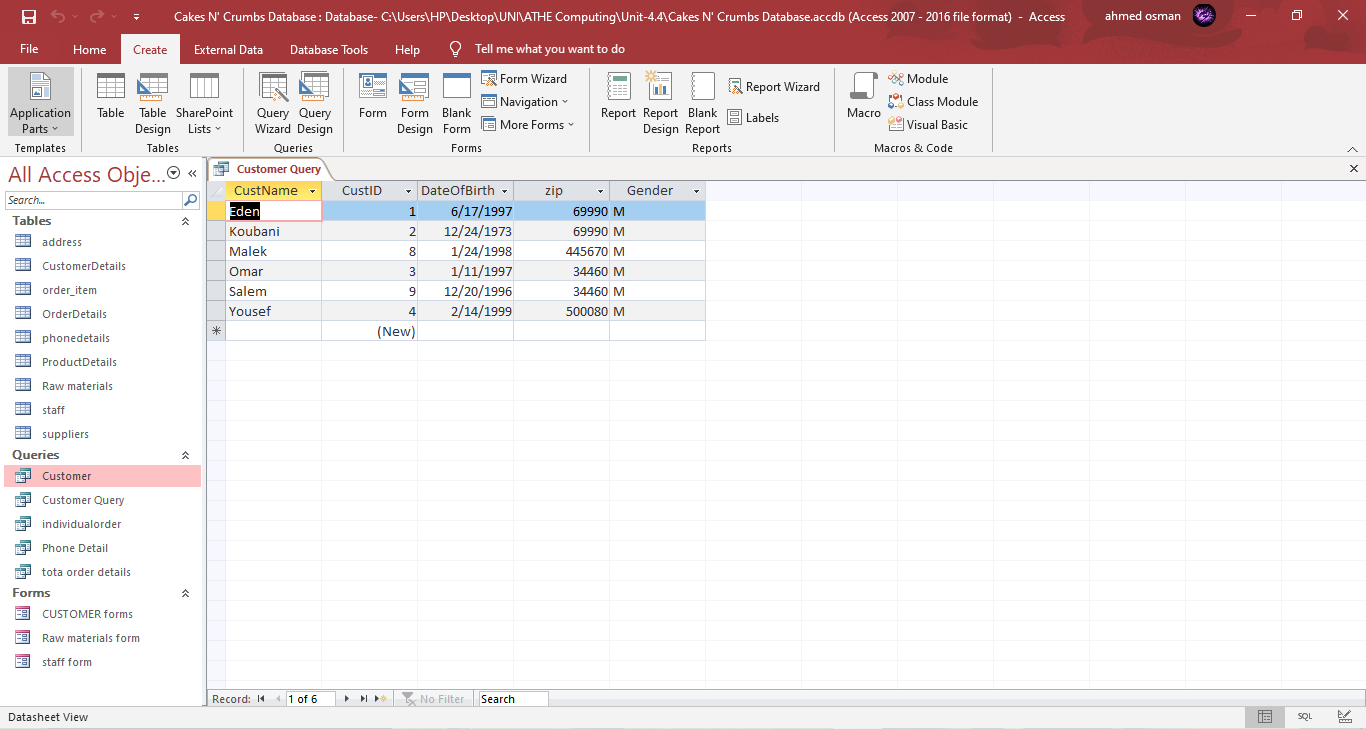


FIG-1.10: Queries Table

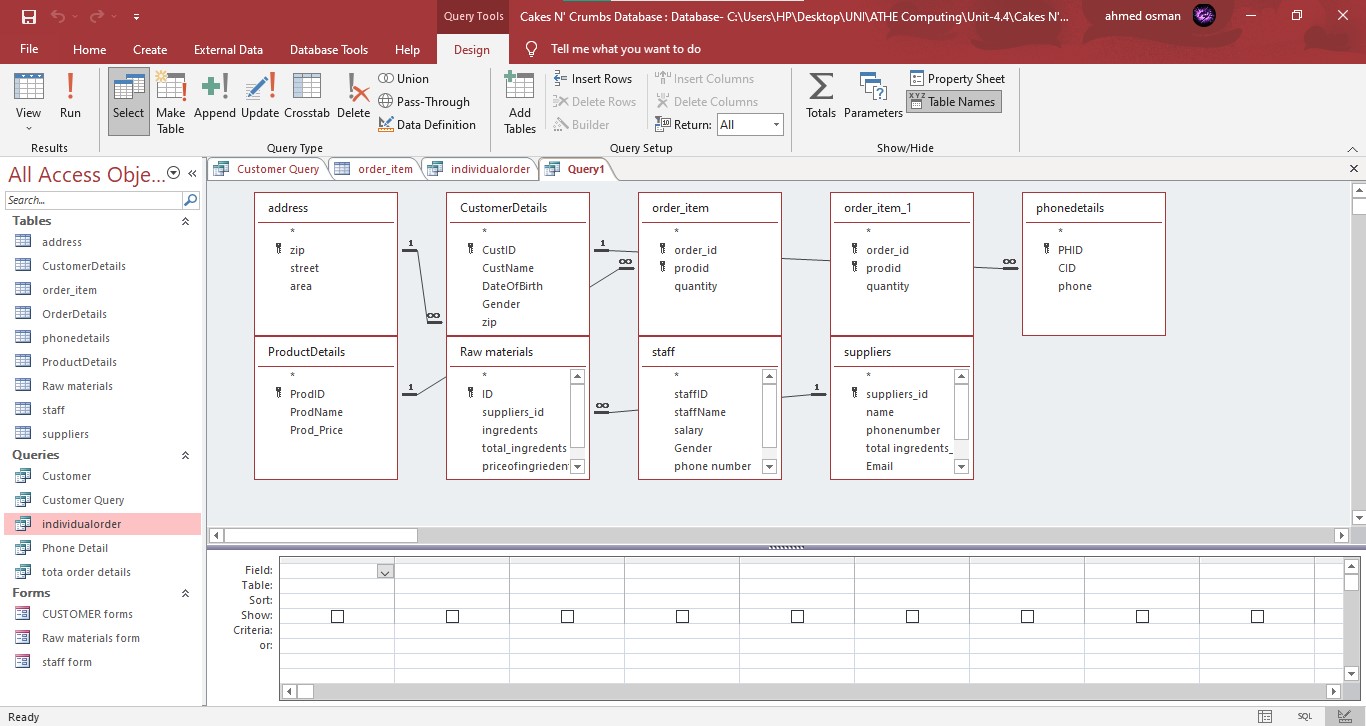


FIG-1.11: Queries 2- Individual Order

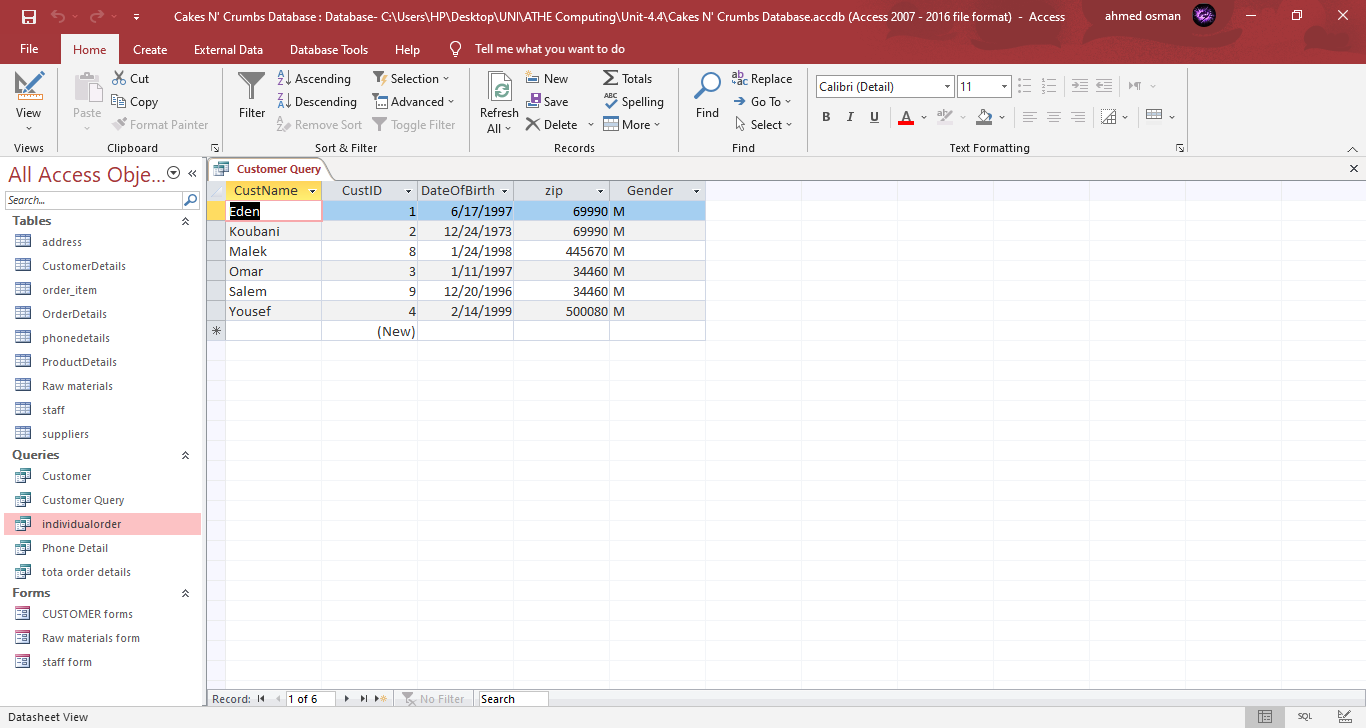


FIG-1.12: Queries Individual Order

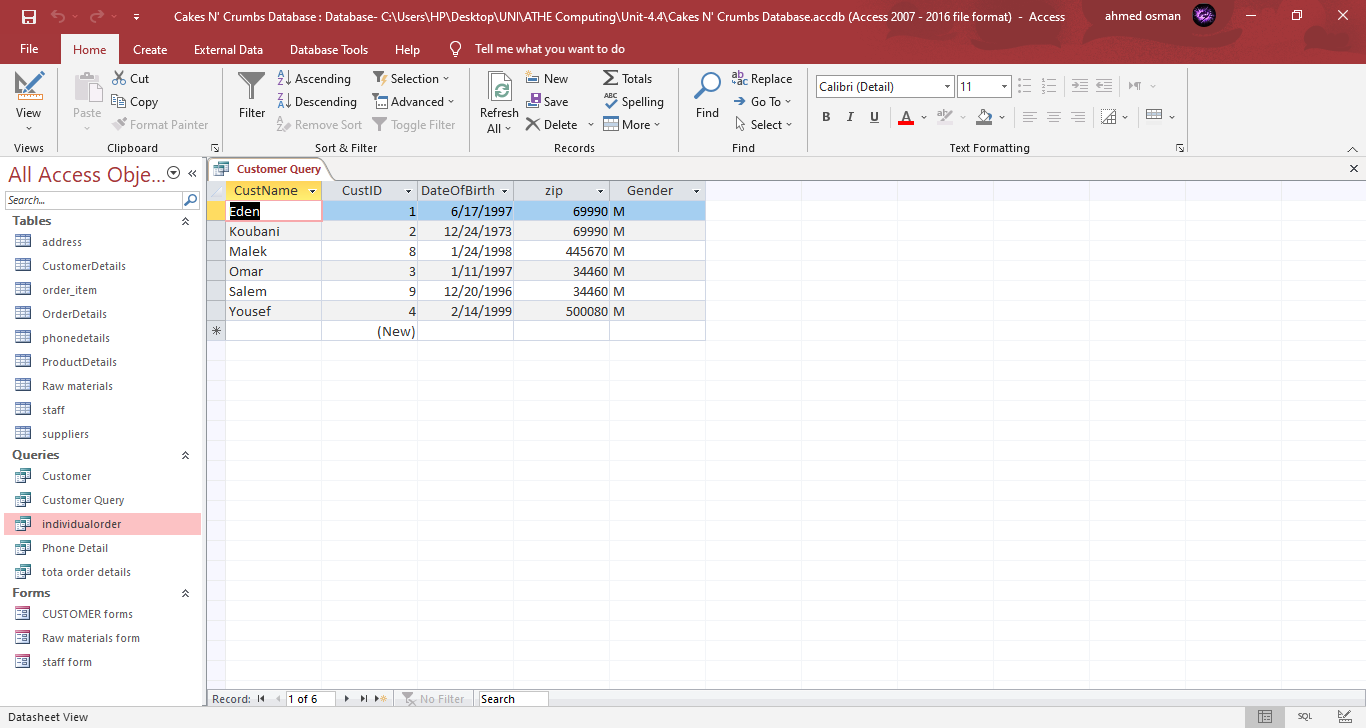


FIG-1.13: Individual order SQL View

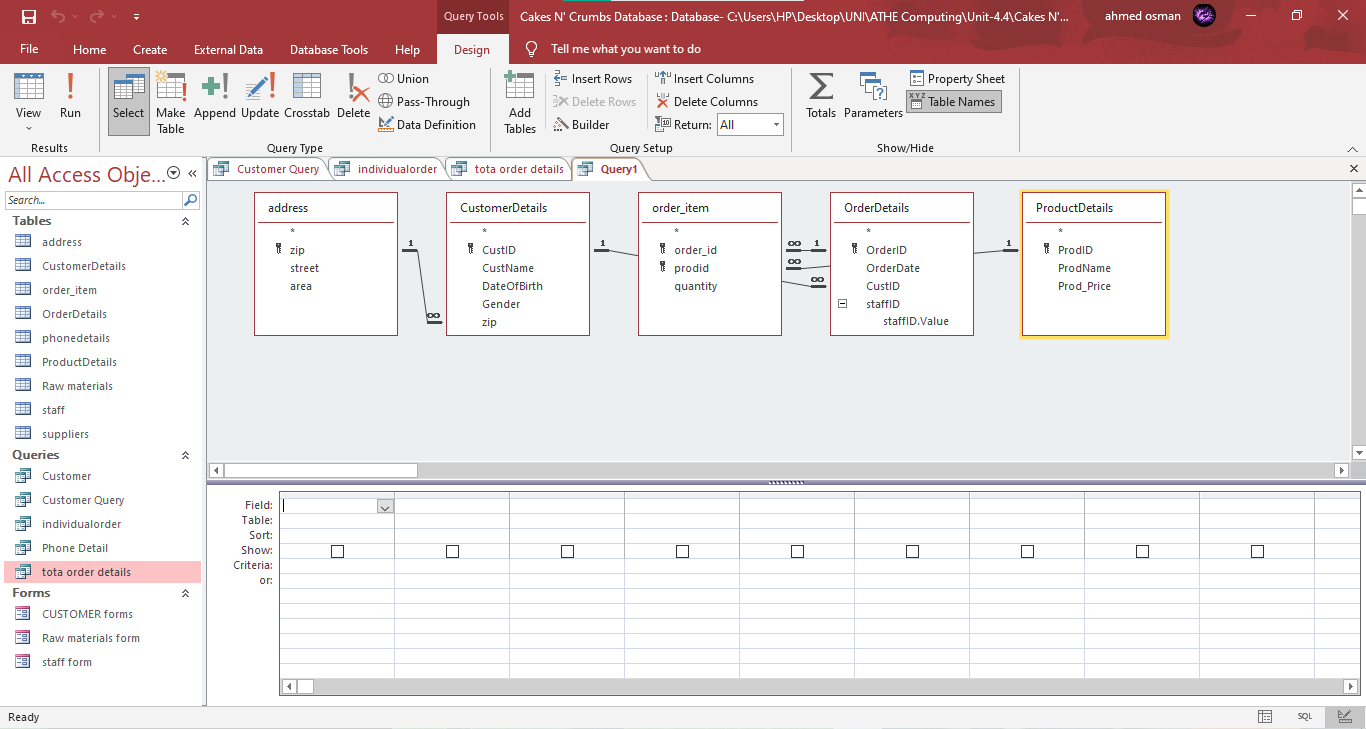


FIG-1.14: Total order table

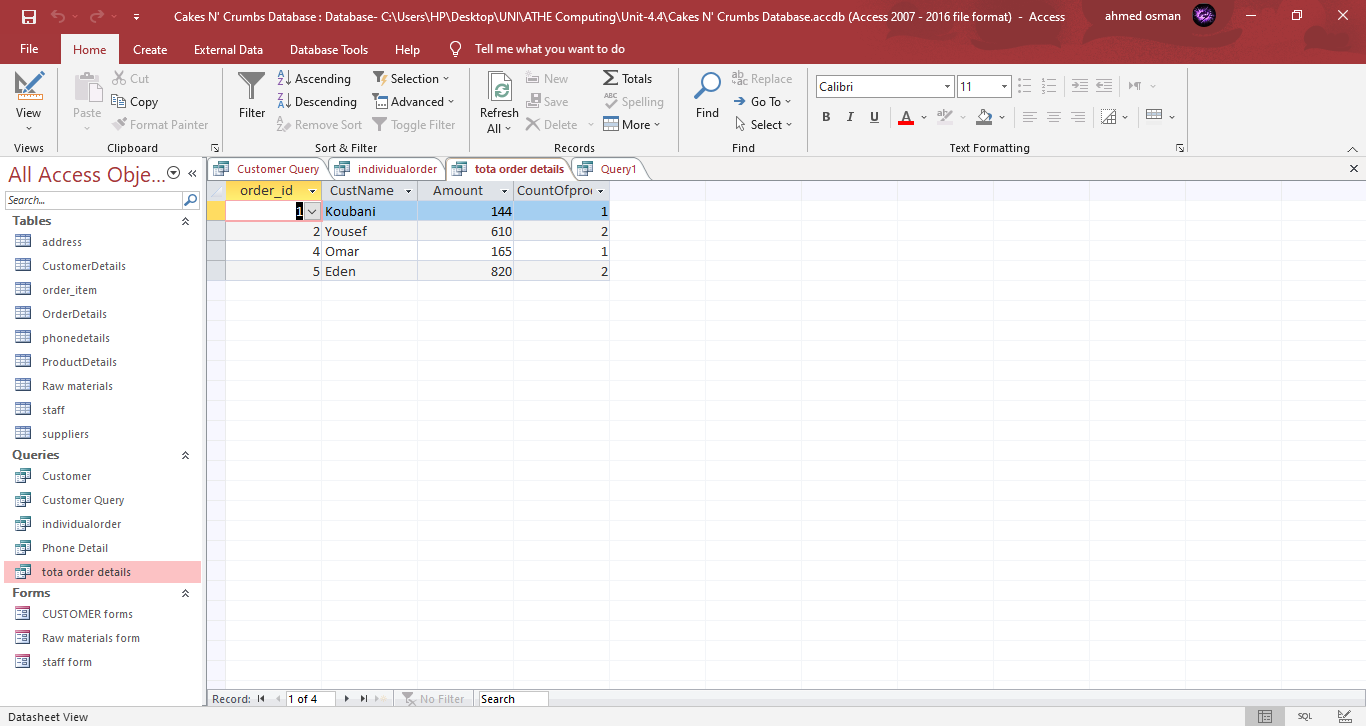


FIG-1.15: Total order queries table

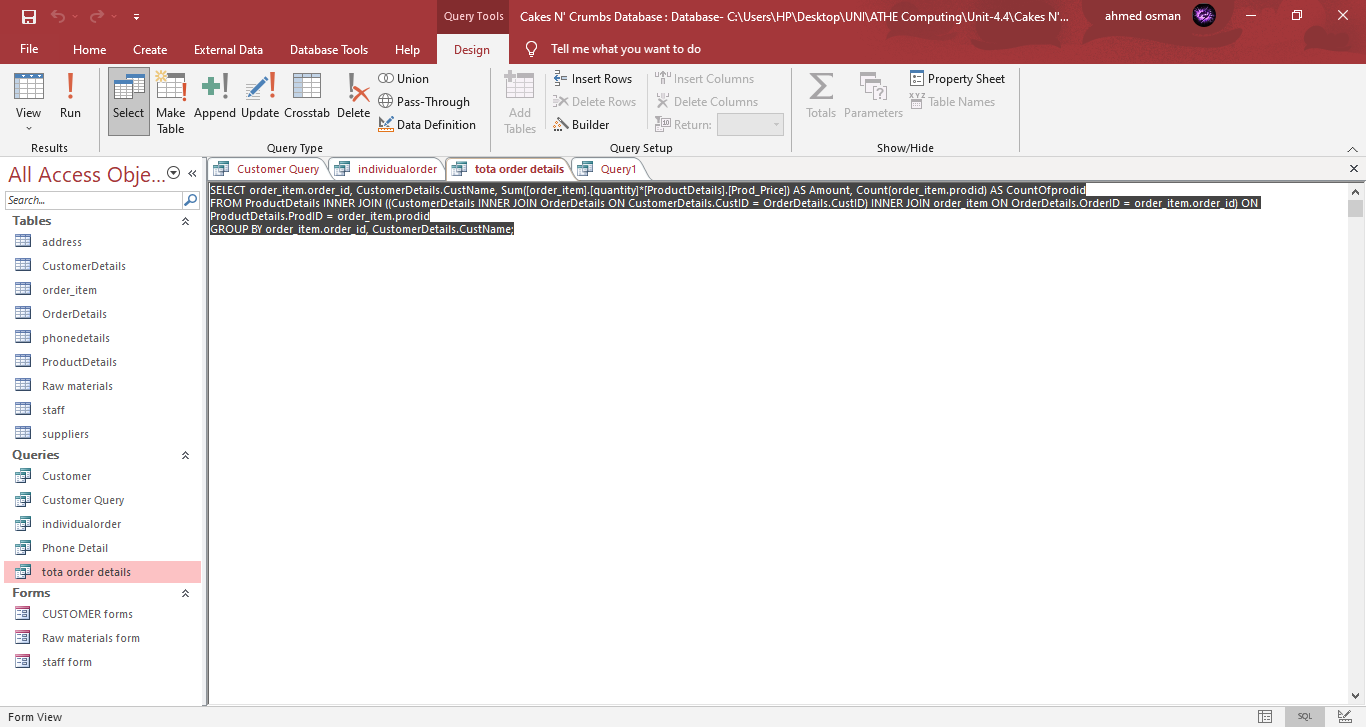


FIG-1.16: Total order SQL view

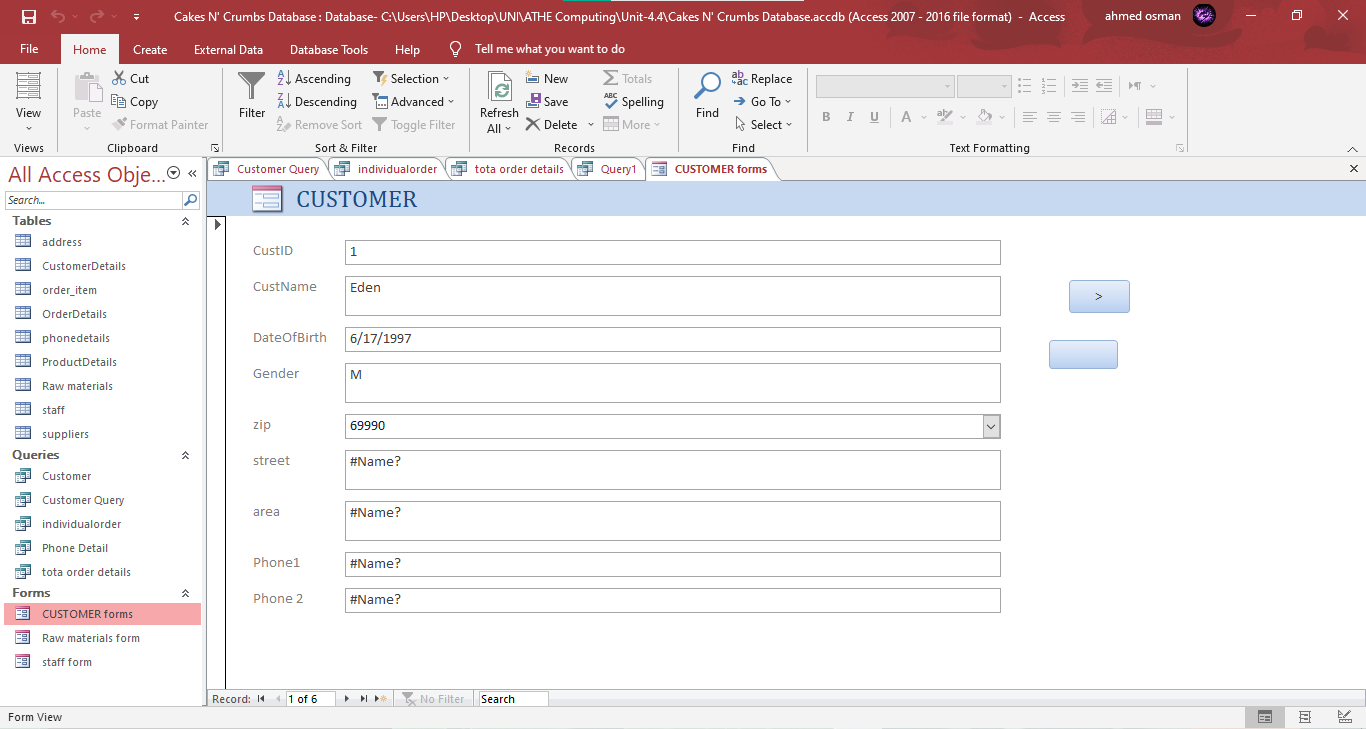


FIG-1.17: Customer Form

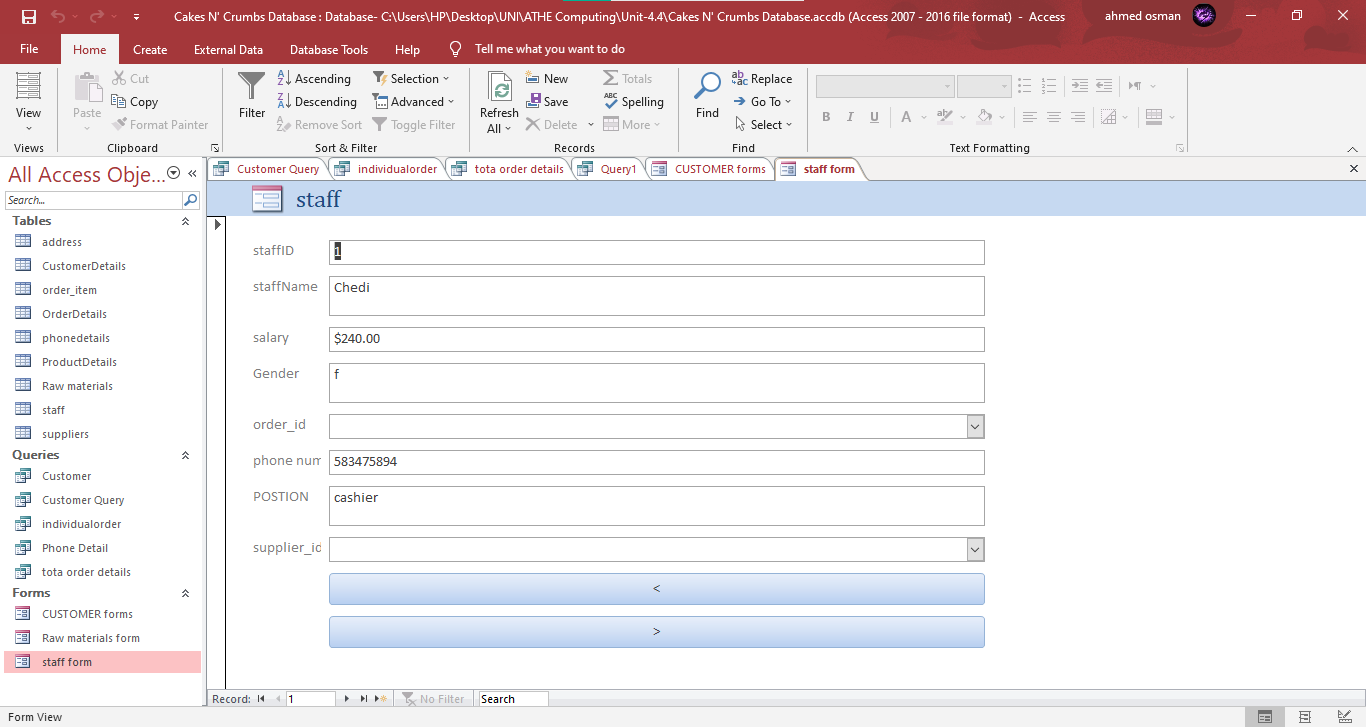


FIG-1.18: Staff Form

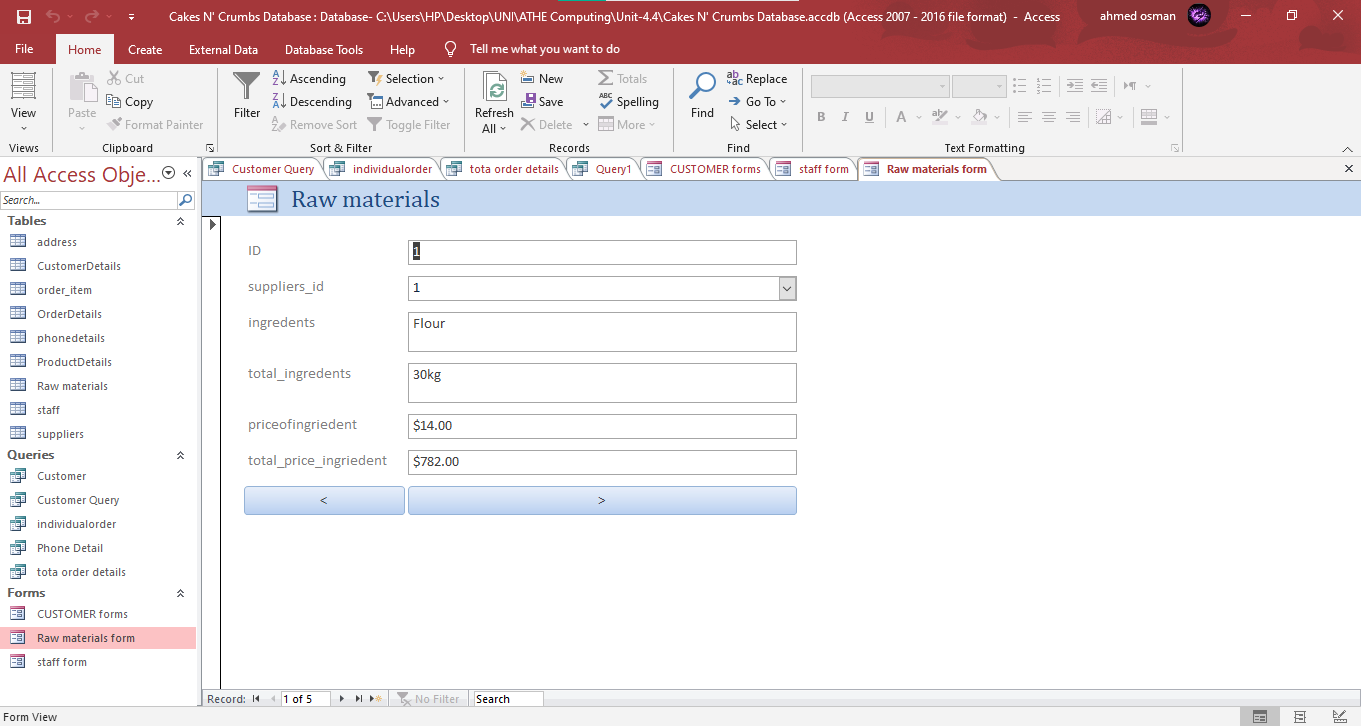


FIG-1.19: Raw materials Form

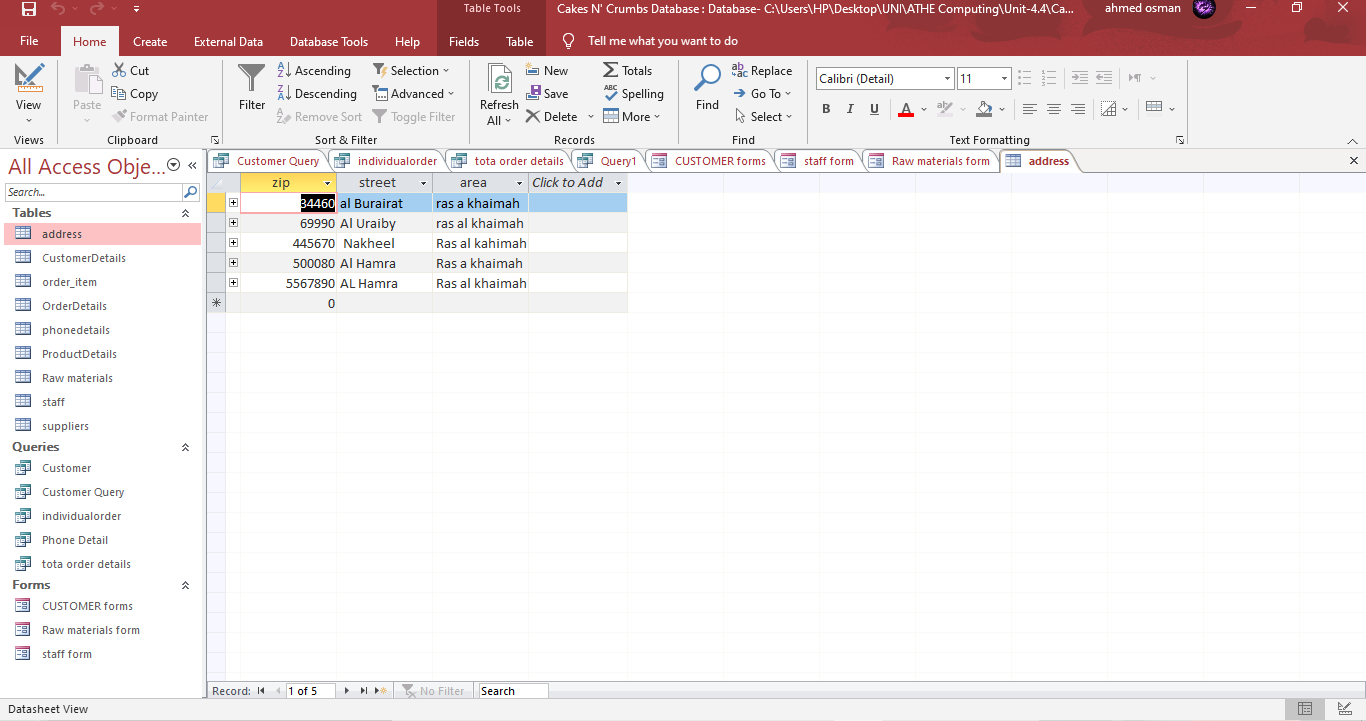


FIG-1.20: Address Table

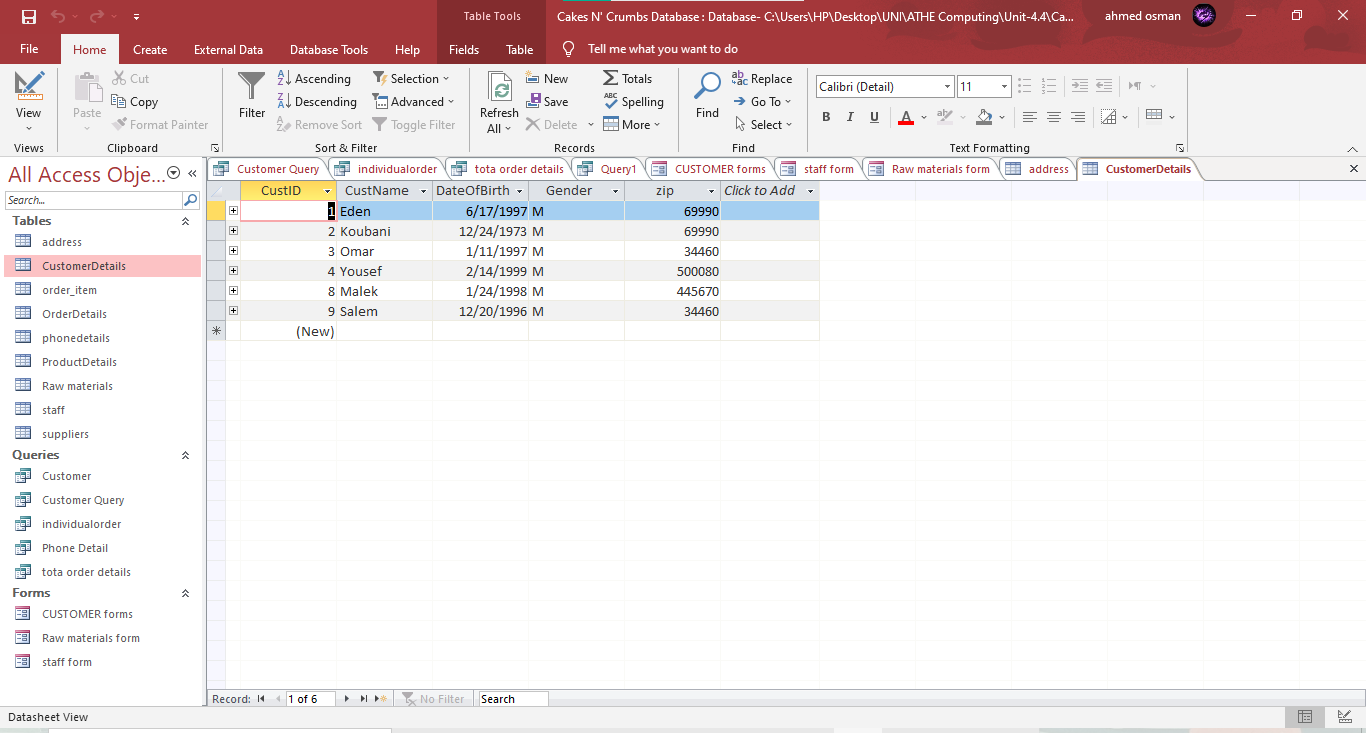
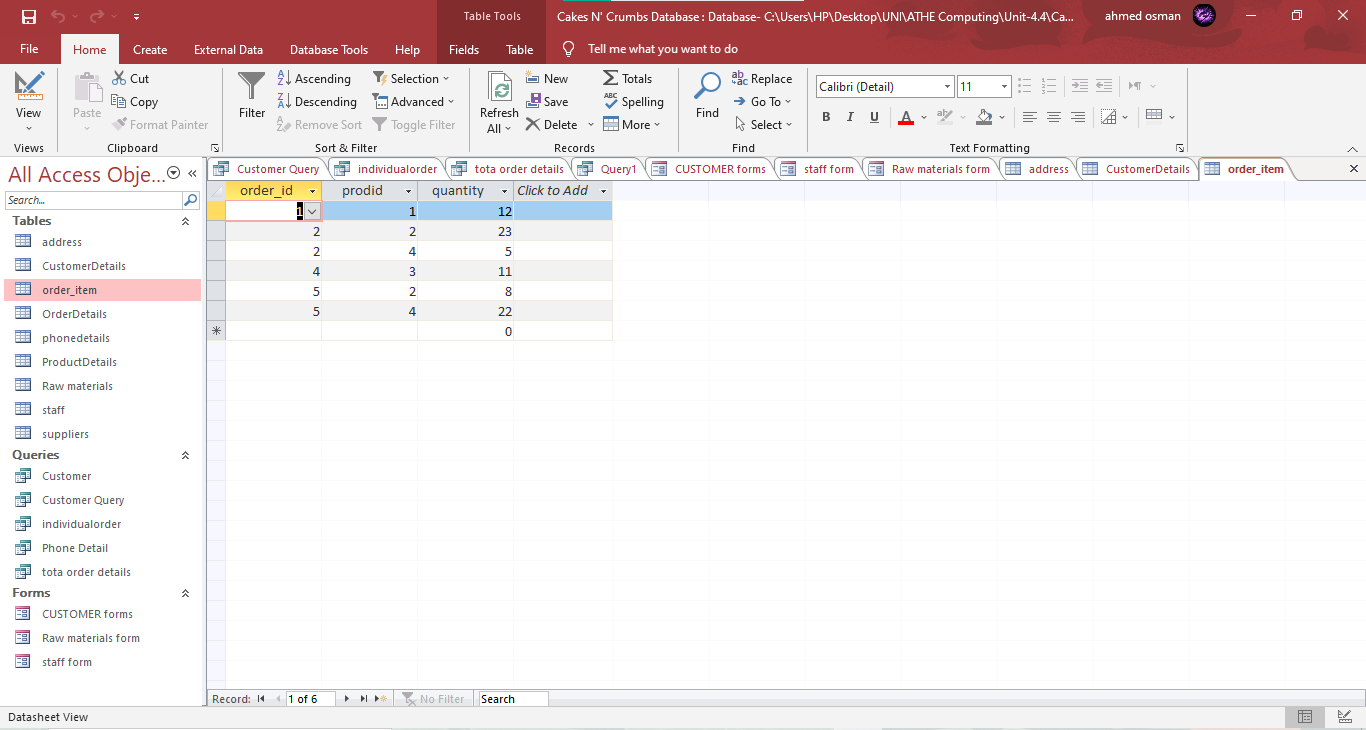
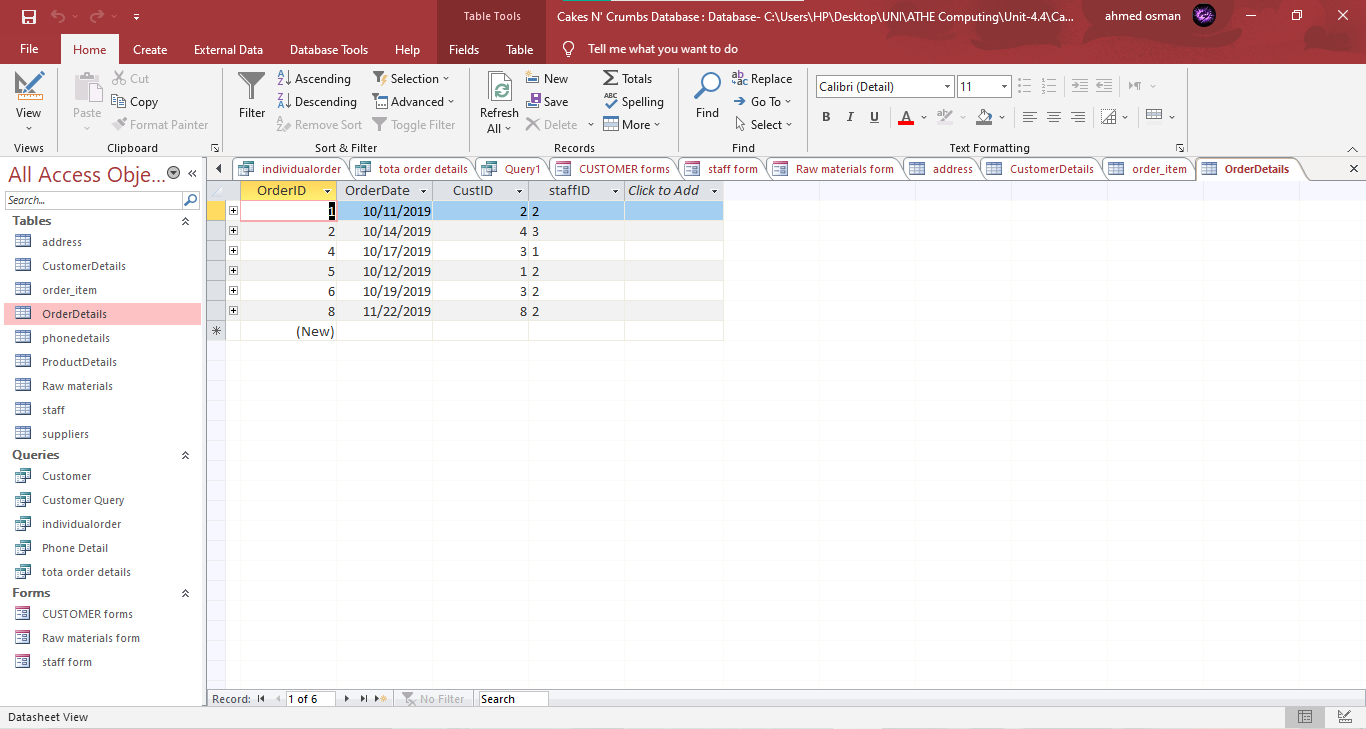


FIG-1.21: Customer details table

FIG-1.22: Order Item table

FIG-1.23: Order Details table

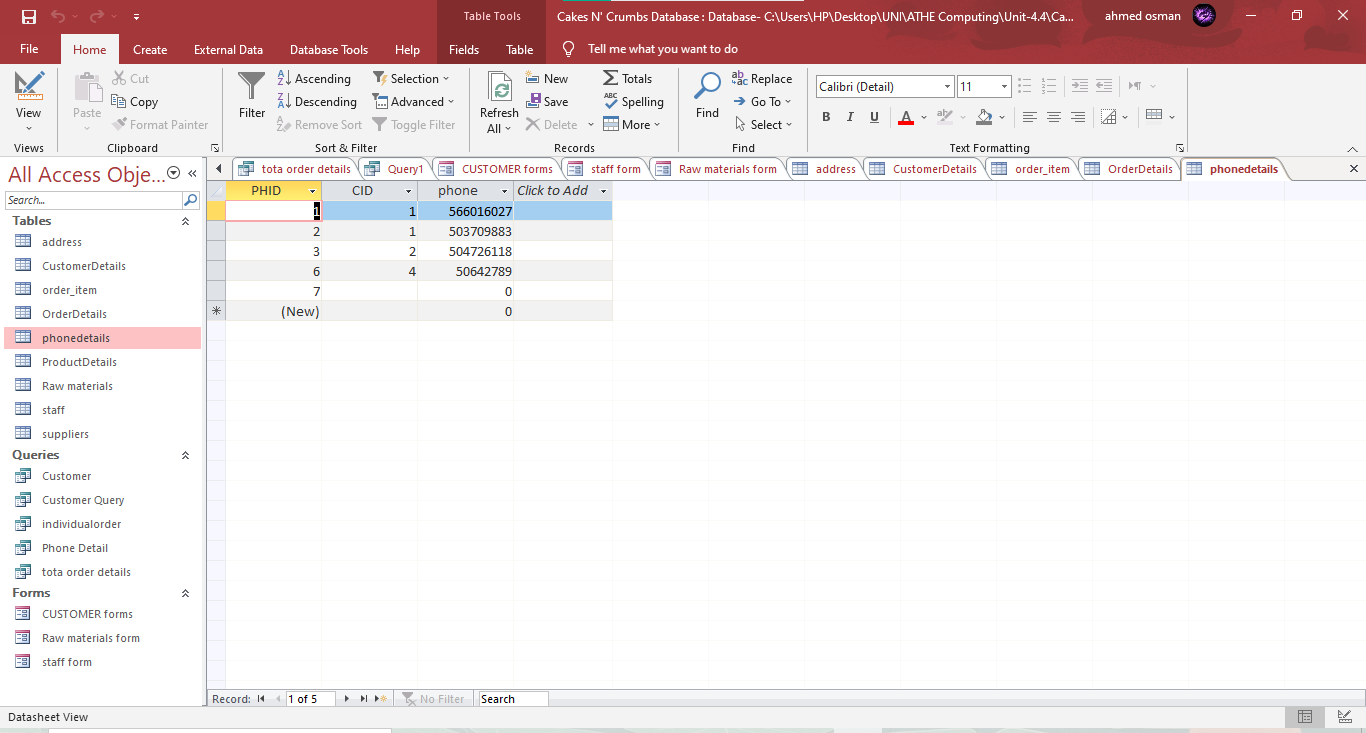


FIG-1.24: Phone Details table

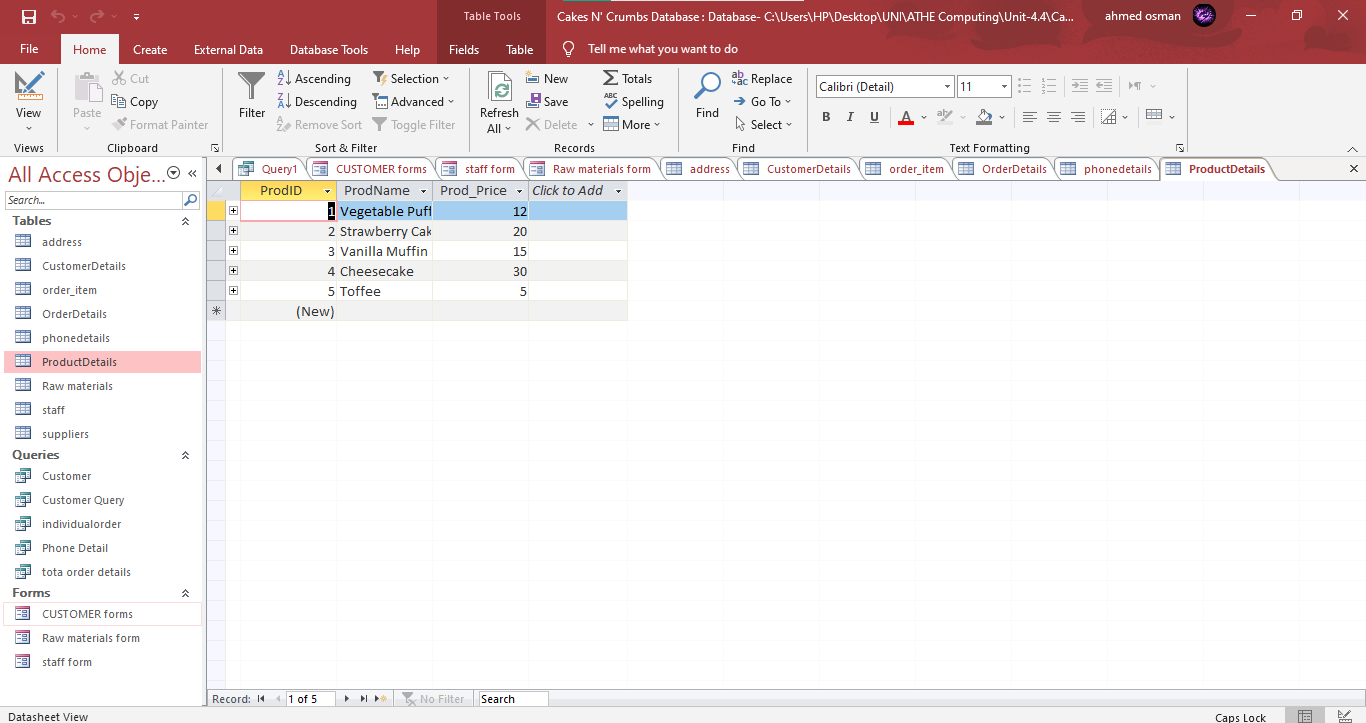


FIG-1.25: Product details table

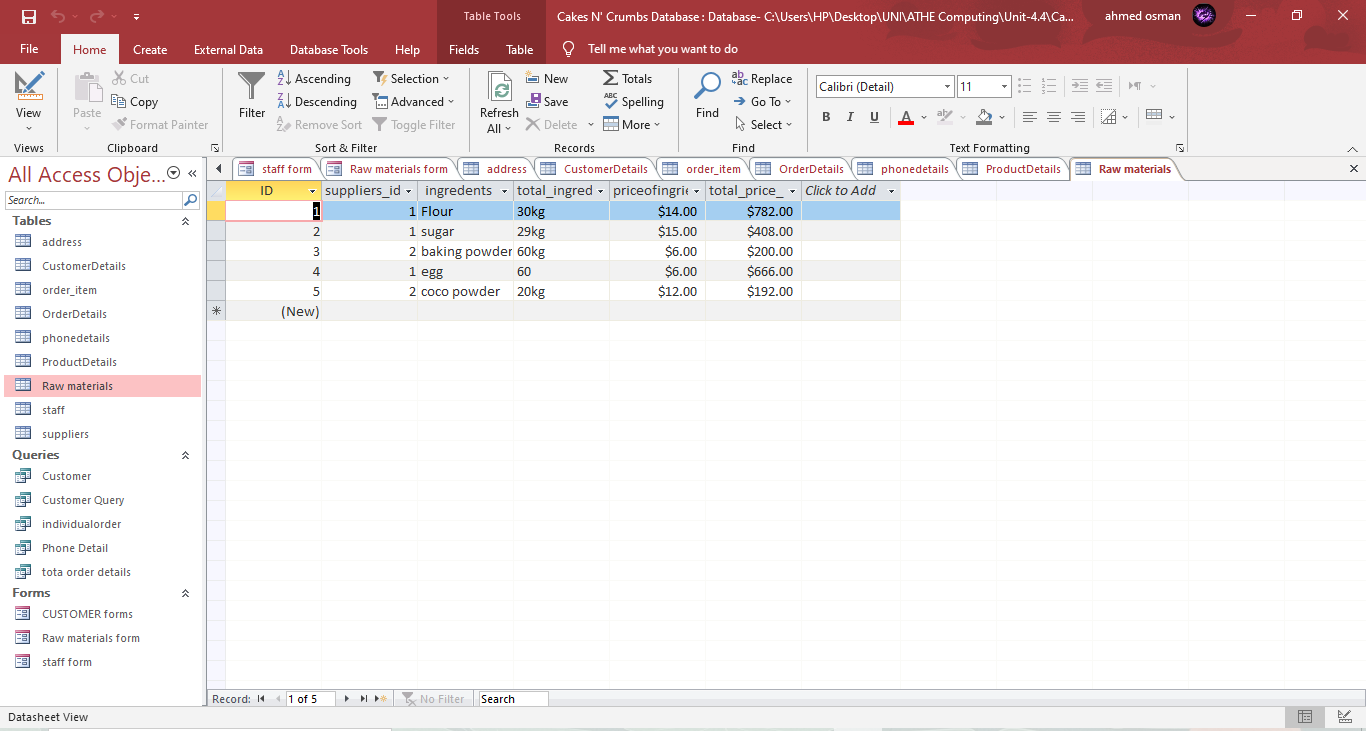


FIG1.26: Raw Materials Table

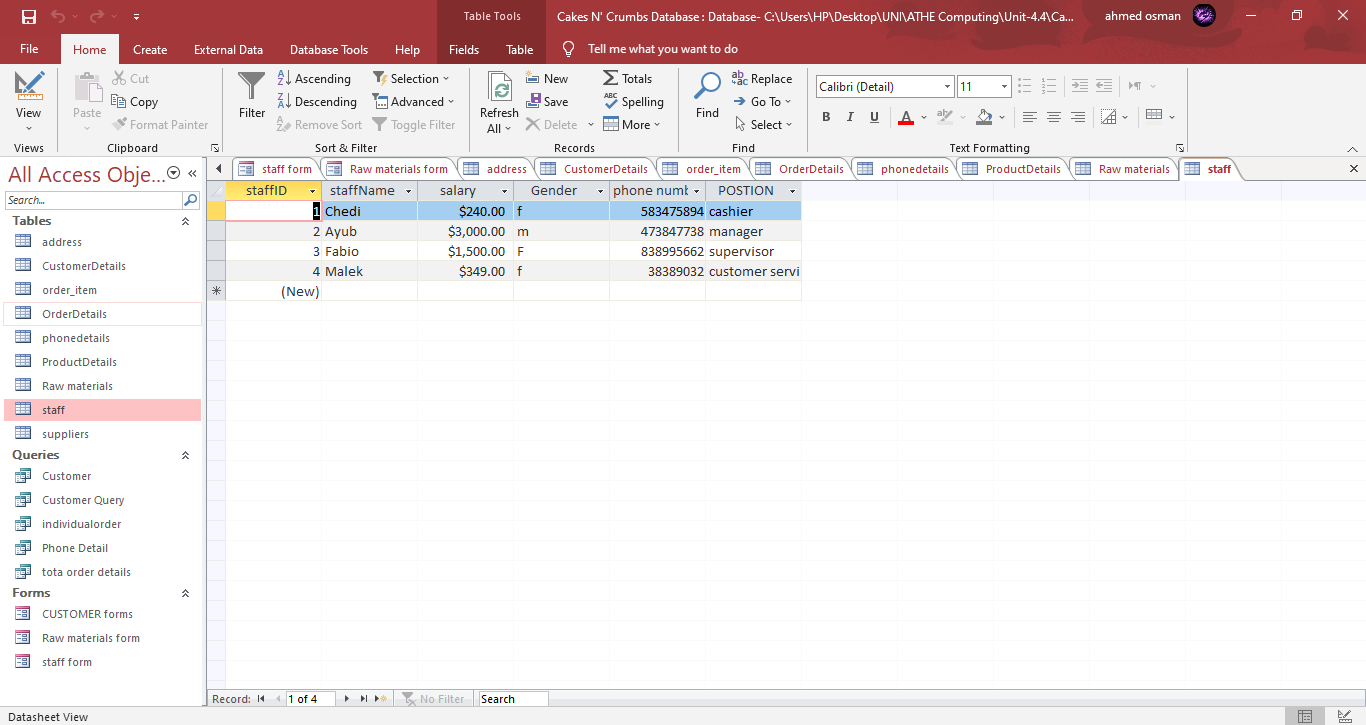


FIG-1.27: Staff table

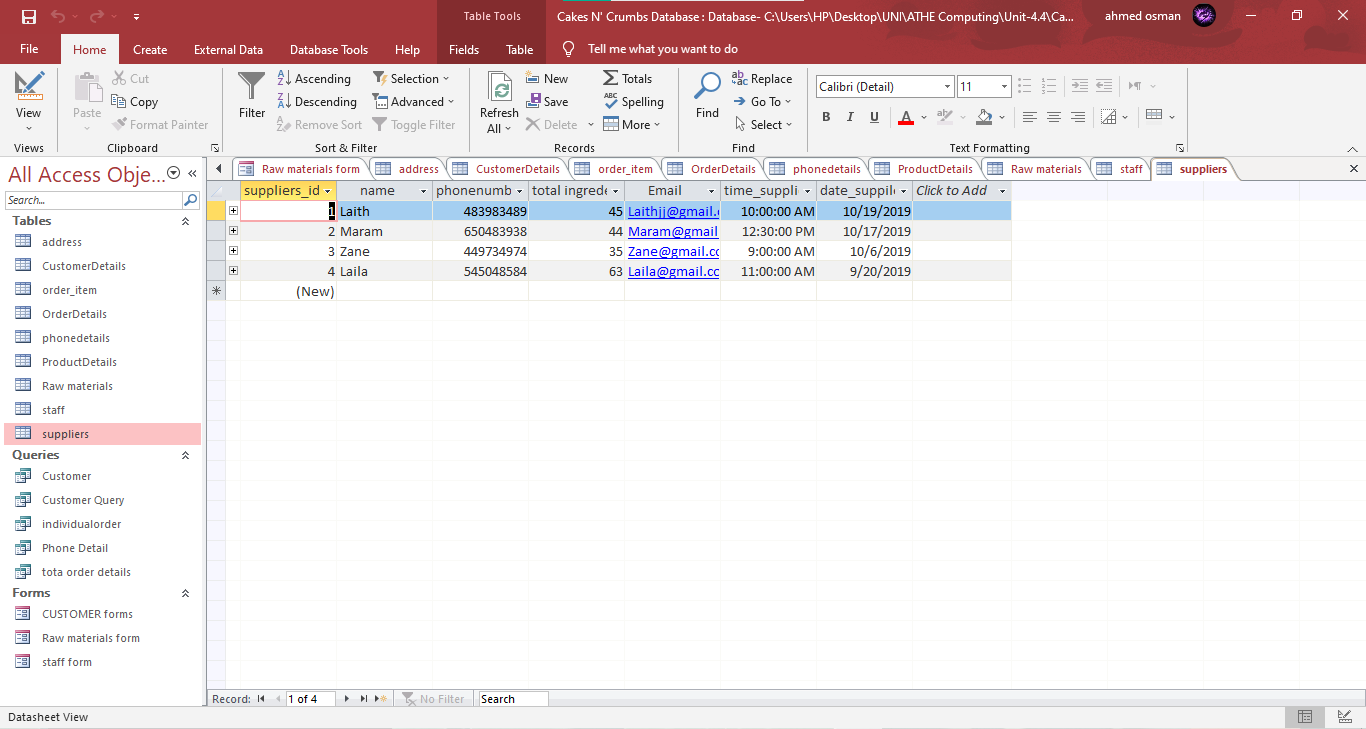


FIG-1.28: Supplier table

# Task-3.f

There were various errors in the entity relation diagram, such as missing entities and improper relationships. As a result of the teachers' feedback, i made adjustments to our ER diagram and our database is now scenario-based.

ERD:. There was no attempt to normalize the data. I had to normalize the data because there were some missing points.

DAC [ GRANT / REVOKE], MAC, RBAC, Different Integrity, Definition of Security, DBA Roles, DAC [ GRANT / REVOKE], DAC [ GRANT / REVOKE], DAC [ GRANT / REVOKE

Put StaffId in its proper place. Table of Contents

For the Scenario provided, perform normalization using the Graphical form.

Which of the database constraints did you create?

GRANT/REVOKE Syntax is missing from a password-protected database snapshot.

Result and SQL queries View and Forms are not available.

These changes were made based on points given by the client

Reference list

Activesocialcare.com. (2021). [online] Available at: https://activesocialcare.com/handbook/handling-information/the-data-protection-act-1998-8-principles [Accessed 5 Apr. 2021].

APQC. (2019). *What are the Best Four Components of Knowledge Management?* [online] Available at: https://www.apqc.org/blog/what-are-best-four-components-knowledge-management [Accessed 7 Apr. 2021].

Css.edu. (2021). *CSSLibraryGuides: EDU 6180 - Understanding & Evaluating EDU Research: Ethical Use of Information*. [online] Available at: https://libguides.css.edu/c.php?g=514195&p=3512710#:~:text=Ethical%20use%20of%20information%20means%20using%20information%20ethically.&text=people’s%20work%20correctly.-,Even%20though%20information%2C%20words%2C%20and%20ideas%20are%20not%20concrete%2C,can%20still%20get%20in%20trouble. [Accessed 5 Apr. 2021].

https://www.facebook.com/marketing91 (2019a). *12 Different Types Of Management Information Systems*. [online] Marketing91. Available at: https://www.marketing91.com/types-of-management-information-system/ [Accessed 8 Apr. 2021].

https://www.facebook.com/marketing91 (2019b). *12 Different Types Of Management Information Systems*. [online] Marketing91. Available at: https://www.marketing91.com/types-of-management-information-system/ [Accessed 8 Apr. 2021].

Investopedia. (2021). *Supply Chain Management (SCM): What You Need to Know*. [online] Available at: https://www.investopedia.com/terms/s/scm.asp#:~:text=Supply%20chain%20management%20is%20the,competitive%20advantage%20in%20the%20marketplace. [Accessed 6 Apr. 2021].

Microsoft Corporation (2021). *What is ERP and why do you need it?| Microsoft Dynamics 365*. [online] Microsoft.com. Available at: https://dynamics.microsoft.com/en-us/erp/what-is-erp/#:~:text=ERP%20is%20an%20acronym%20that,manufacturing%2C%20and%20human%20resource%20activities. [Accessed 6 Apr. 2021].

Mindtools.com. (2017). *Porter’s Five Forces: - Understanding Competitive Forces to Maximize Profitability*. [online] Available at: https://www.mindtools.com/pages/article/newTMC\_08.htm [Accessed 6 Apr. 2021].

NACDL - National Association of Criminal Defense Lawyers. (2021). *NACDL - Computer Fraud and Abuse Act (CFAA)*. [online] Available at: https://www.nacdl.org/Landing/ComputerFraudandAbuseAct [Accessed 5 Apr. 2021].

OAIC. (2015). *The Privacy Act*. [online] Available at: https://www.oaic.gov.au/privacy/the-privacy-act/ [Accessed 5 Apr. 2021].

Really Simple Systems. (2021). *What is Sales Force Automation? | Really Simple Systems CRM*. [online] Available at: https://www.reallysimplesystems.com/blog/sales-force-automation/ [Accessed 6 Apr. 2021].

Robb, D. (2015). *10 Data Storage Tips | How to Improve Your Storage | ESF*. [online] Enterprise Storage Forum. Available at: https://www.enterprisestorageforum.com/management/data-storage-tips/ [Accessed 13 Jun. 2021].

Salesforce.com. (2019). *What is CRM?* [online] Available at: https://www.salesforce.com/crm/what-is-crm/ [Accessed 6 Apr. 2021].

Sciencedirect.com. (2016). *Computer Security Act - an overview | ScienceDirect Topics*. [online] Available at: https://www.sciencedirect.com/topics/computer-science/computer-security-act [Accessed 5 Apr. 2021].

serverlost uniten. (2015). *4 GENERIC STRATEGIES FOR DEALING WITH COMPETITIVE FORCE,ENABLE BY USING IT*. [online] Available at: https://serverlost.wordpress.com/2015/04/27/4-generic-strategies-for-dealing-with-competitive-forceenable-by-using-it/ [Accessed 6 Apr. 2021].

State.gov. (2021a). *The Freedom of Information Act*. [online] Available at: https://foia.state.gov/learn/foia.aspx#:~:text=generally%20provides%20that%20any%20person,special%20law%20enforcement%20record%20exclusions. [Accessed 6 Apr. 2021].

State.gov. (2021b). *The Freedom of Information Act*. [online] Available at: https://foia.state.gov/learn/foia.aspx#:~:text=generally%20provides%20that%20any%20person,special%20law%20enforcement%20record%20exclusions. [Accessed 6 Apr. 2021].

Techterms.com. (2020). *Copyright*. [online] Available at: https://techterms.com/definition/copyright [Accessed 5 Apr. 2021].

The Business Professor, LLC. (2021). *Structured Problem vs Unstructured Problems - Explained*. [online] Available at: https://thebusinessprofessor.com/management-leadership-organizational-behavior/structured-problem-vs-unstructured-problem [Accessed 8 Apr. 2021].

Usability.gov. (2021). *User Interface Design Basics | Usability.gov*. [online] Available at: https://www.usability.gov/what-and-why/user-interface-design.html [Accessed 13 Jun. 2021].