## **UNIT 4.4**

## **RELATIONAL DATABASE SYSTEMS**

**TUTOR NAME: MS. IBTISAM MOGUL**

**Submitted by: Fatima Ejaz Barri**

Contents

[**UNIT 4.4** 1](#_Toc79610061)

[**RELATIONAL DATABASE SYSTEMS** 1](#_Toc79610062)

[LO1- 4](#_Toc79610063)

[1.1 explain the database management system (DBMS) 4](#_Toc79610064)

[What is database 4](#_Toc79610065)

[What is database management system? 4](#_Toc79610066)

[Advantages of database management system 5](#_Toc79610067)

[Controlling data redundancy: 5](#_Toc79610068)

[Sharing of data: 5](#_Toc79610069)

[Data consistency: 5](#_Toc79610070)

[Integration of data 5](#_Toc79610071)

[Integration constraints 5](#_Toc79610072)

[Data security 5](#_Toc79610073)

[Report writers 5](#_Toc79610074)

[Backup and recovery procedures 5](#_Toc79610075)

[Control over concurrency 5](#_Toc79610076)

[Data independence 6](#_Toc79610077)

[APPLICATIONS OF DATABASE MANAGEMENT SYSTEM 6](#_Toc79610078)

[Banking system: 6](#_Toc79610079)

[Telecom: 6](#_Toc79610080)

[Airlines: 6](#_Toc79610081)

[Manufacturing: 6](#_Toc79610082)

[Educational: 6](#_Toc79610083)

[Online shopping: 6](#_Toc79610084)

[API example: 7](#_Toc79610085)

[1.2 explain the different levels of database architecture 7](#_Toc79610086)

[The three levels of Database architecture 7](#_Toc79610087)

[What is big data? Data: 7](#_Toc79610088)

[Social media 8](#_Toc79610089)

[Types of big data 8](#_Toc79610090)

[Structured 8](#_Toc79610091)

[Unstructured: 9](#_Toc79610092)

[Semi-structured 9](#_Toc79610093)

[3vs of data big 10](#_Toc79610094)

[Velocity: 11](#_Toc79610095)

[Applications of big data 11](#_Toc79610096)

[1.4 what is transaction processing? 11](#_Toc79610097)

[4properties of transactions 11](#_Toc79610098)

[Task 2 12](#_Toc79610099)

[2. A) Entity Relationship diagram – ERD 12](#_Toc79610100)

[Make an entity diagram a bakery management system 14](#_Toc79610101)

[THE DETAILS OF YOUR ENTITIES IN THE DATABASE 14](#_Toc79610102)

[Reference 23](#_Toc79610103)

## LO1-

## 1. A) an explanation of the database management system (DBMS)

## What is database

A database is an organized, consisted and logical collection of knowledge which can easily be updated, accessed, and managed. The data stored in a database can be of any type like student database, employee database, hospital database etc.

## Database management system (DBMS)

The Database management system is a software which is used to manage the database. It is a software system that uses a standard method of cataloguing, retrieving and running queries on data. The DBMS manages incoming data, organizes it, and provides ways for the data to be modified or extracted by users or other programs. For example: include Microsoft SQL server, dBase, MySQL, Oracle database, Microsoft Access and FileMaker pro etc. are a very popular commercial database which is used in different applications. DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.

## Applications of Database management system

Banking system:

For customer information, bank transaction, loans and accounts.

Telecom:

For keeping records of calls made, network usage, customer details and storing information about the communication network.

Airlines:

For reservations and schedule information. Airline were among the first to use the database in a geographically disturbed manner-terminals situated around the word accessed the central database system through phone lines and other data networks.

Manufacturing:

For management of supply chain for tracking the production of items in factories, inventories of items in warehouses

Educational:

Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees detail etc. there is a huge amount of interrelated data which is processed quite efficiently.

Online shopping:

You must be aware of online shopping websites such as amazon, flipchart etc. These sites store the product information, your addresses and preference, credit card details (credit details means your balance) and provide you with the relevant list of products based on your query. All this involves a database management system.

## b) Explain the different levels of database architecture

There are three levels of the database architecture following:

External level:

Internal level:

This level is made out of the inside blu

Conceptual level:

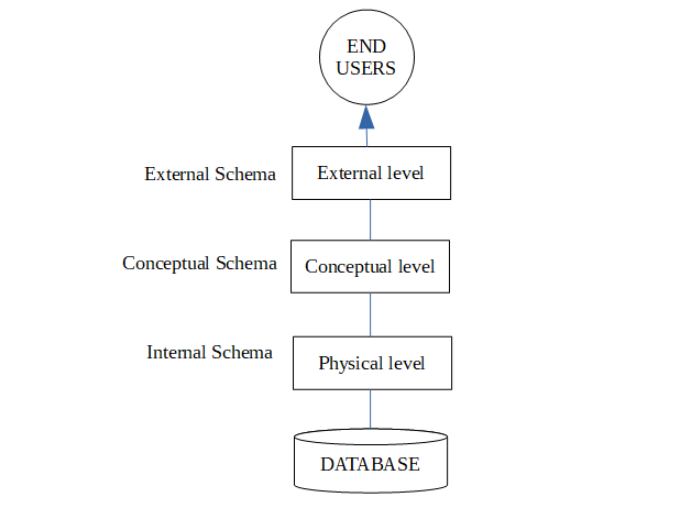


Fig: 1.1- 3 level of architecture of DBMS. (tutorialspoint.com).

## 1.3 c) a description of big data and how it can apply to the database management system

## What is big data?

The numbers, characters or symbols on which a device conducts operations that can be processed and distributed in the form of electrical signals, and recorded on magnetic, optical, or mechanical storage devices.

Big data is also data but it has an enormous size. Big data is a term used to describe a collection of data that is small in scale which expands exponentially over time. In short, this data is so large and complex that none of the conventional data management systems can efficiently store or process it.

# 3vs of data big

The following 3 dimensions are commonly used for more precise. These dimensions are generally referred to as 3vs of big data:

## Velocity:

Velocity describes the frequency of generation, capture and sharing of the data. Recent Development that’s more data is generated in much shorter cycles not only by consumers but also by businesses. This data can only be capitalized on by velocity companies if the data is collected and transmitted in real- time.

## Variety:

As many sources are contributing to big data, the type of data they are generating is different. It is often structured, semi-structured or unstructured.

Volume:

This is the incredible amount of knowledge that’s generated from different sources like social media, sensors and public releases within the sort of public data, photographs, and videos. That is so large that it cannot be stored using techniques to store and analyse data.

## Applications of big data

Big Data is the term used to refer to big amounts of complicated data and it is used to make it easier for people to access and analyse the data of big corporations. This is the basis of the working of many new industries. Following are some of the fields where big data is used:

Ecommerce:

Online business alludes to the utilization of electronic means and advancements to direct trade (deals, buy, move or trade of items, administration as well as data), including inside business, business-to-business and business-to-customer collaborations.

Healthcare

In healthcare, big data can be used to make the healthcare experience more customized and help the patient access information about the clinic and its employees.

Social media

With the increase in the use of social media, the data about every single user has to be stored and processed which requires a lot of energy and time. Along these lines, huge information is an answer for the issue. All social media apps use big data to store the immense amount of information it gathers from its user’s activities.

Government

The government has a variety of sectors with different types of information stored, it could be about the citizens, the government officials, foreign relations, etc. All this data is composed and stored with the help of big data.

1. D) an explanation of transaction processing within a database management system

What is transaction processing?  
Transaction processing is the division of data processing into different steps and these small steps are called transactions. All these steps are interrelated and the next cannot be initiated until the previous transaction has been completed or failed. This helps to make the processing easier.

## Transaction Properties (ACID)

A transaction (database operations) could be standard specified in application-oriented language like SQL submitted interactively, or could be embedded within a program (Java, Python or C++). There is set of properties that guarantee that Data base transactions are performed through following ACID properties.

Atomicity:

A transaction is an atomic unit of processing; it should either be performed in its entirety or not performed at all. There must be no state in a database where a transaction is left partially completed. States should be defined either before the execution of the transaction or after the execution/abortion/failure of the transaction.

For example, in an application that transfers funds from one account to another, the atomicity property ensures that, if a debit is made successfully from account, the corresponding credit is made to the other account execution/abortion/failure of the transaction.

Consistency:

The database must remain in a consistent state after any transaction.

No transaction should have any adverse effects on the data residing in the database. If the database was in a consistent state before the execution of transaction, it must remain consistent after the execution of the transaction as well.

Isolation:

It is a property that ensures that one operation cannot affect others. To maintain consistency in a database, it follows AICD properties. Among these four properties (atomicity, consistency, isolation and durability) isolation means that a transaction should take place in a system in such way that it is the only transaction that is accessing the resources in a database system. This ensures that two transactions on the same information are independent and do not generate any type of error.

Durability requirement:

In database systems, durability is the ACID property which ensures that once the operation has been carried out, it will persist and cannot be undone even if the system fails.

## Transaction states

Each transaction has the following 5 states:

Active:

Partially committed:

Committed:

Failed:

Aborted:

## Transaction manager

## OLTP OLAP

The work of prior on-line functional frameworks was to perform exchange and question processing. Along these lines, they are likewise named as on-line transaction processing frameworks (OLTP). Information distribution centre frameworks serve clients or information labourers in the job of information investigation and dynamic. Such frameworks can put together and present information in different designs to oblige the different necessities of the various clients. These frameworks are approached line logical preparing (OLAP) frameworks.

## 1. E) an evolution of the importance of data integrity and quality control within a management system

## GIGO

It stands for “Garbage In, Garbage Out.” GIGO is a computer science acronym that implies bad input will result in bad output. Because computers operate using strict logic, invalid input may produce unrecognizable output, or “garbage”. For example, if a program asks for an integer and you enter a string, you may get an unexpected result.

## Data Integrity definition

Data integrity refers to the overall accuracy, completeness, and reliability of data. It can be specified by the lack of data. It can be specified by the lack of variation between two instances or consecutive updates of a record, indicating that your information is error-free. It also corresponds to the security of data pertaining to regulatory compliance.

## Integrity Importance

The importance of data integrity in protecting yourself from data loss or a data leak cannot be overstated: in order to keep your data safe outside forces with malicious intent, you must first ensure that internal users are handling data correctly. By implementing the appropriate data validation and error checking, you can ensure that sensitive data is never miscategorised or stored incorrectly, thus exposing you to potential risk.

## Data Quality

Data quality ensures that the data stored in your database is compliant with the organisation’s standards and requirements. In other words, it ensures maintaining data integrity in a database. In doing so, it applies a set of rules to a specific or complete dataset and stores it in the targets database.

## Task 2- DATABASE DESIGN

## A) ERD and Data dictionary

## Entity relationship Diagram – ERD

To place it in straightforward terms, an entity-relationship outline otherwise called an ERD is a method that is usually utilized by computer programmers to address information in a manner that would make fathoming it simpler and henceforth make the product plan and creation simpler. An ERD is mostly entrusted with the job of giving the various substance and showing the connection between them such that anyone can comprehend.

There are 4 basics of ERDS.

* Entity
* Attributes
* Relationship
* Cardinality

Entity

An entity is a thing indistinguishable from all other objects. It can be merely identifiable as a rectangle in an ERD. All entities have some properties or information that define them.

E.g.:

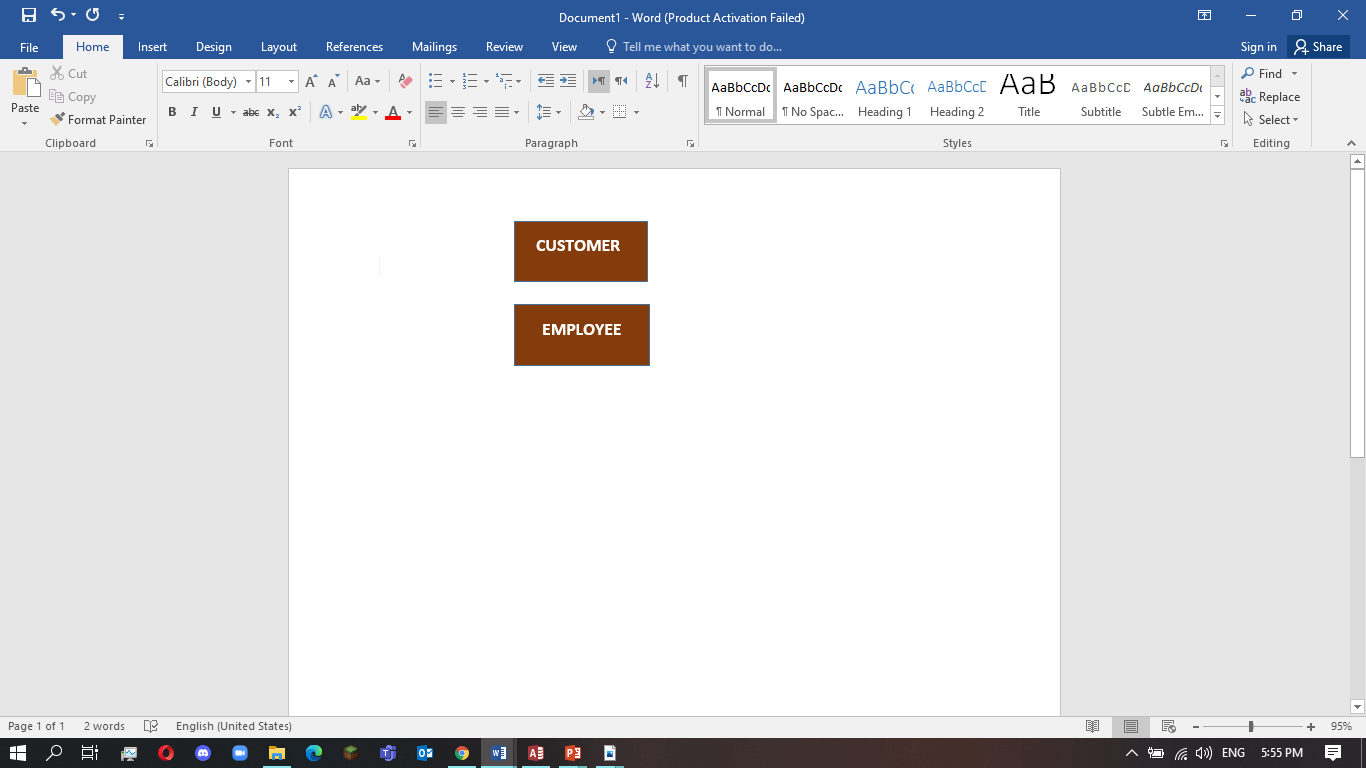


Table- 2.1 entity

Entity set

An entity set may be a set of same sort of entities. They should be interrelating. For e.g.: a customer having a bakery account at a bakery. The customer, employee and department are entities, which are relating and thus make an entity set.

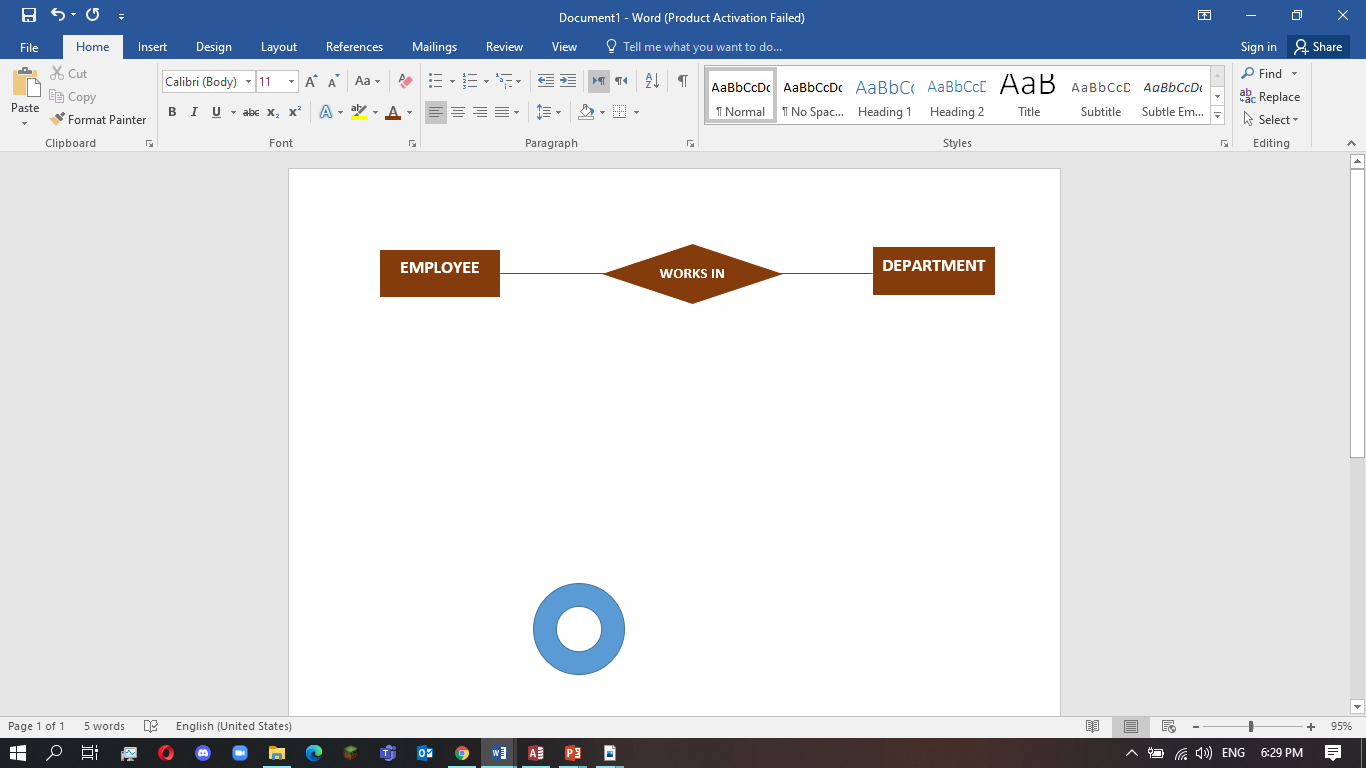


TABLE- 2.2: entity set

Attributes

The information or properties, which define an entity are called attributes of that entity.

E.g. a customer address can have the attributes, street, area and zip.

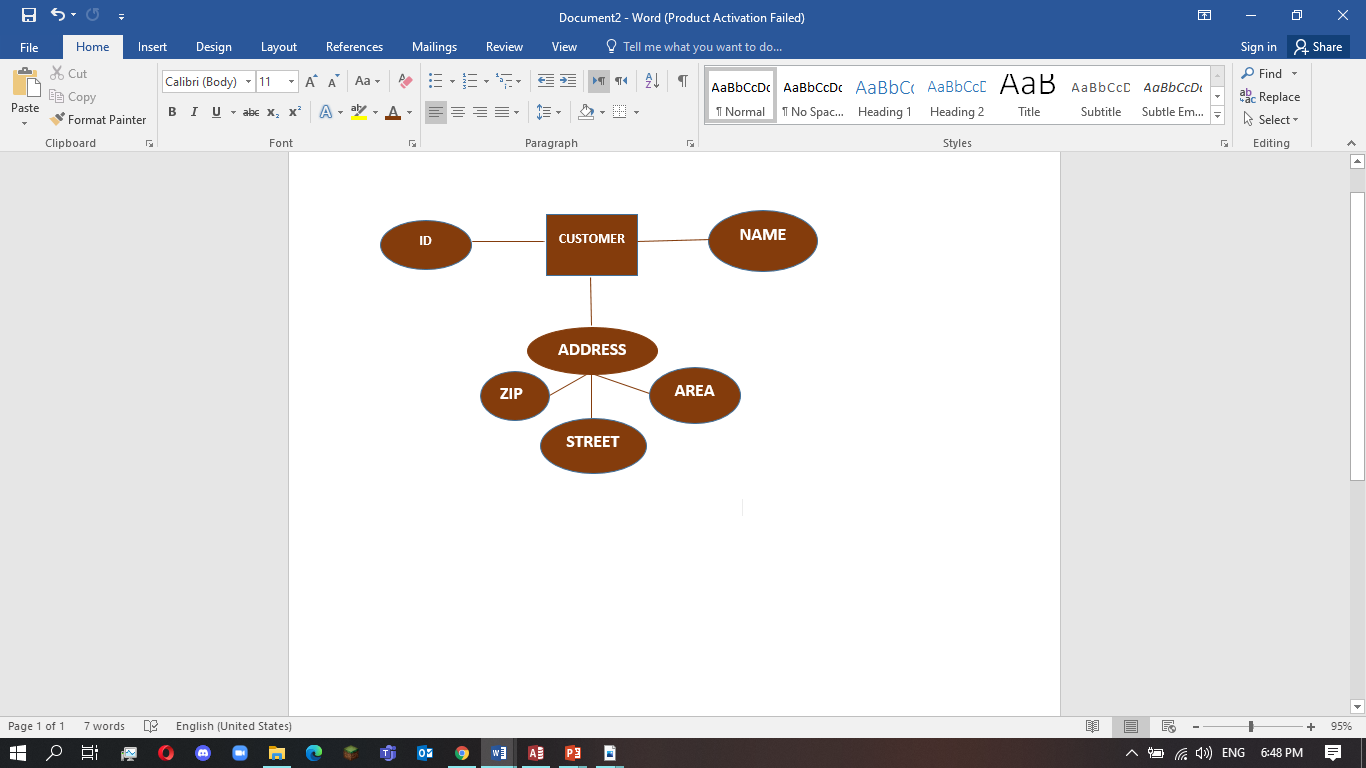


Table: 2.3- attributes

## Identifying entities and their attributes

There are 6 numbers entities in the database.

Entity 1 (Customer)

CustID: the customer identifying of the customer. And it shows the primary key.

CustName: it is the attribute to display the customer name.

DateOfBirth: it is the attribute to display the date of birth of the customer.

Entity 2 (Raw material)

Raw\_materialID: the materials ID identifying of the raw. And it shows the primary key.

Material: it is the attribute to display the material name. And it shows the foreign key.

Price: it is attribute to display the material price.

Entity 3 (suppliers)

Supplier ID: the supplier identifying of the supplier ID.

Name: it is the attribute to display the supplier name.

Zip code: postal code for the supplier.

Entity 4 (Products)

ProdID: the product identifying of the product. And it shows the primary key.

ProdName: it is attribute to display the product name.

ProdPrice: it is attribute to display the product price.

Entity 5 (Orders)

OrderID: the order ID identifying of the order. And it shows the primary key.

OrderDate: it is attribute to display the order date.

Entity 6 (Staff)

Staff ID: the staff identifying of the staff ID. And it shows the primary key.

Staff Name: it is attribute to display the staff name.

Staff Contact: contact of the staff.

## Mapping cardinalities

One customer can place many orders.

One order can include quantity many products

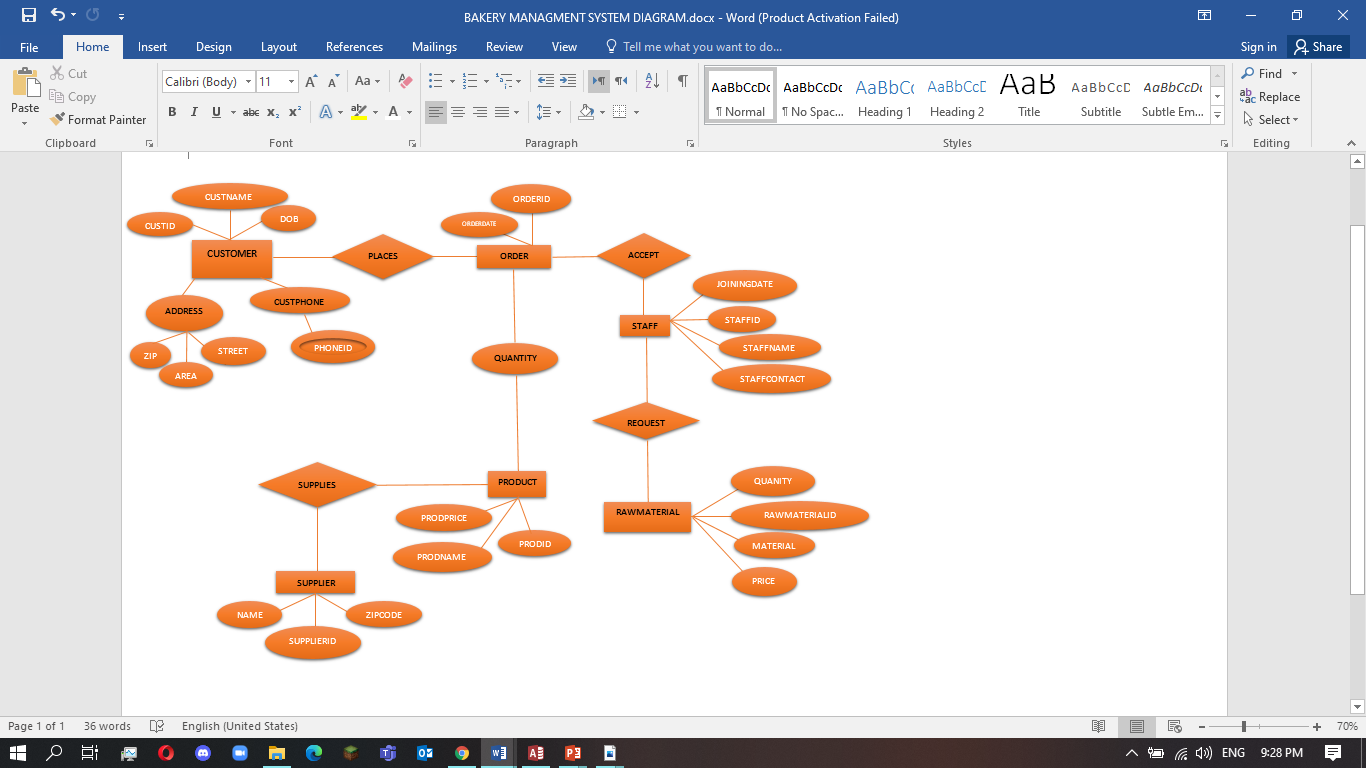
One product can supply many supplier

One supplies supplier one product.

One raw material makes only 1 product is made only 1 product.

One product is made by only 1 staff.

# Make an entity diagram a bakery management system



1. Data dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 1 Customer** | | | | |
| **Field** | **Optional** | **Format** | **Length Limit** | **Description** |
| CustID | N | NUMBER | - | Primary key |
| CustName | N | TEXT | 50 | - |
| DateOfBirth | Y | DATE | - | - |
| Gender | Y | MALE/FEMALE | - | - |
| Zip | N | NUMBER | - | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 2 Suppliers** | | | | |
| **Field** | **Optional** | **Format** | **Length Limit** | **Description** |
| SupplierID | N | AUTO NUMBER | - | Primary key |
| Name | N | TEXT | 50 | - |
| ZipCode | N | NUMBER | - | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 3 Raw Materials** | | | | |
| **Field** | **Optional** | **Format** | **Length Limit** | **Description** |
| Raw\_Materials | N | AUTO NUMBER | - | Primary key |
| Materials | N | TEXT | 50 | Foreign key |
| Price | N | NUMBER | - | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 4 Products** | | | | |
| **Field** | **Optional** | **Format** | **Length Limit** | **Description** |
| ProdID | N | AUTO NUMBER | - | Primary key |
| ProdName | N | TEXT NUMBER | 50 |  |
| Prod\_Price | N | NUMBER | - | - |

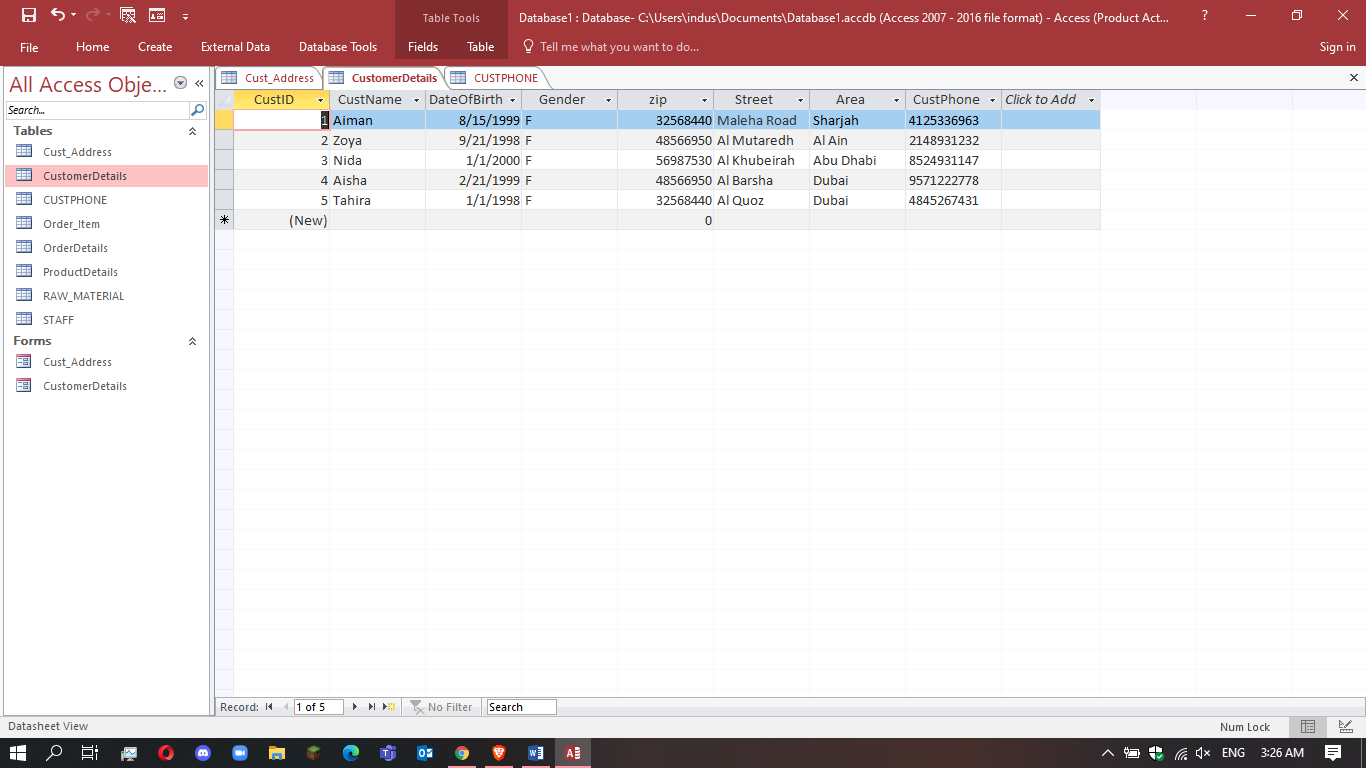
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 5 Orders** | | | | |
| **Field** | **Optional** | **Format** | **Length** | **Description** |
| OrderID | N | AUTO NUMBER | - | Primary key |
| OrderDate | N | DATE | - | - |
| CustID | N | NUMBER | - | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 6 Staff** | | | | |
| **Field** | **Optional** | **Format** | **Length** | **Description** |
| Staff\_ID | N | AUTO NUMBER | - | Primary key |
| Staff\_Name | N | TEXT NUMBER | 50 | - |
| Staff\_Contact | N | NUMBER | - | - |

## 2. B) Normalization

## 1st Normal Form

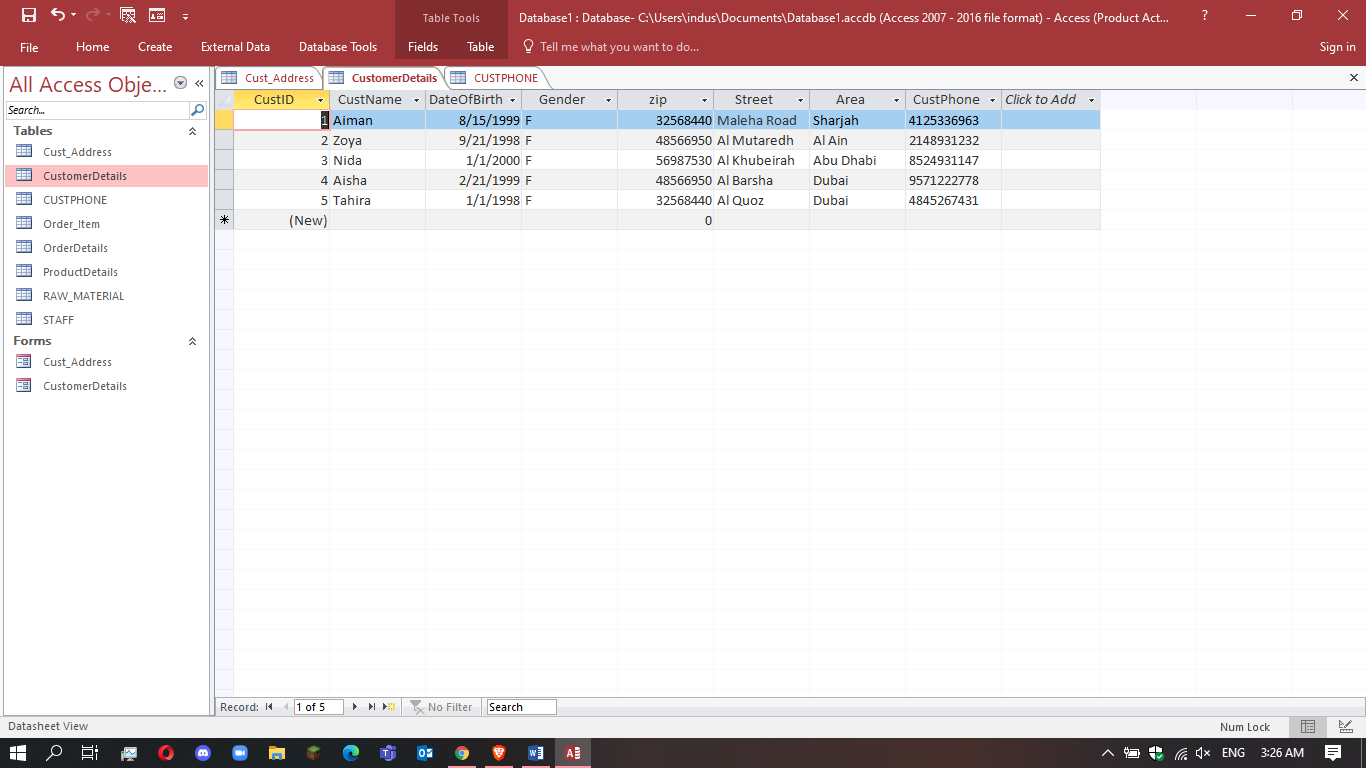
**Consider Customer Table,**



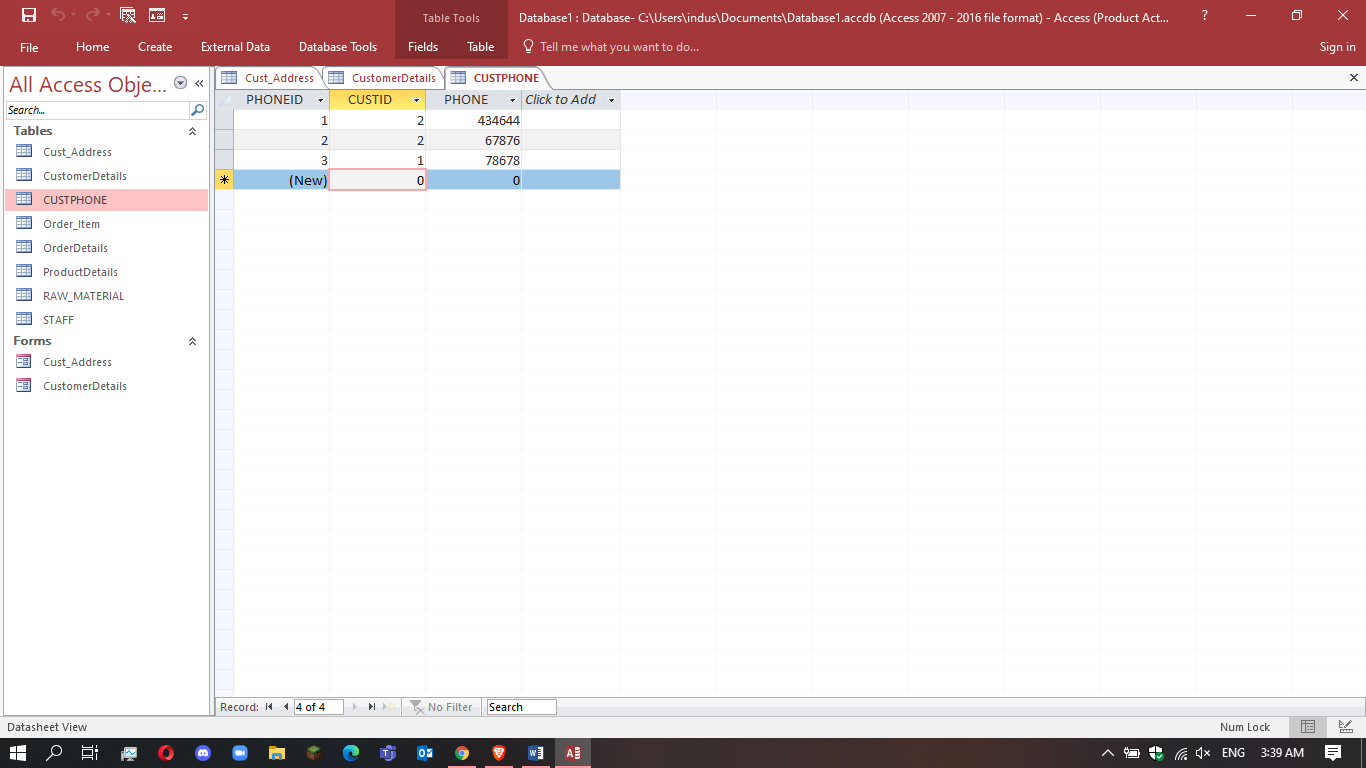
Customer table was not in INF as it had multivalued attribute-CustPhone

So it decomposes into customer table and Customer Phone.

**Customer Table**



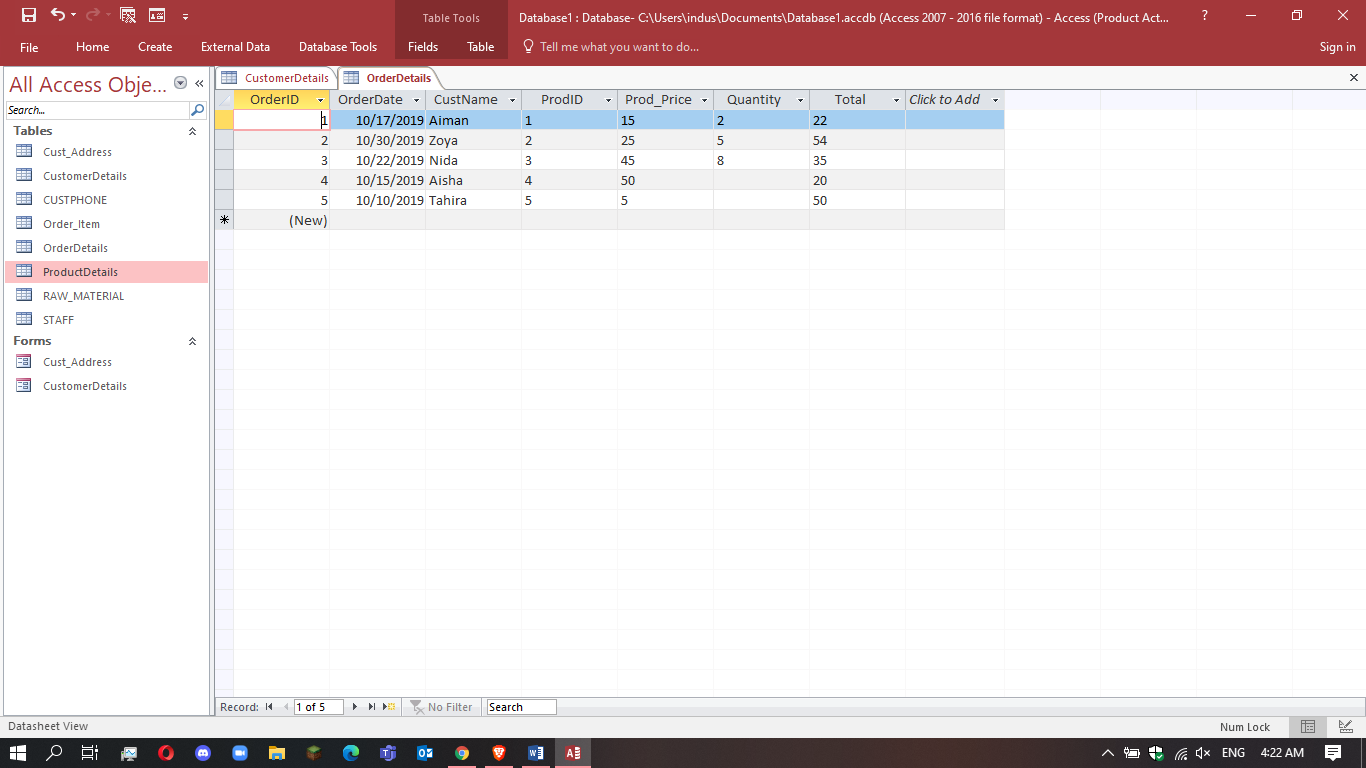
**Customer Phone Table**



Now both tables are in INF.

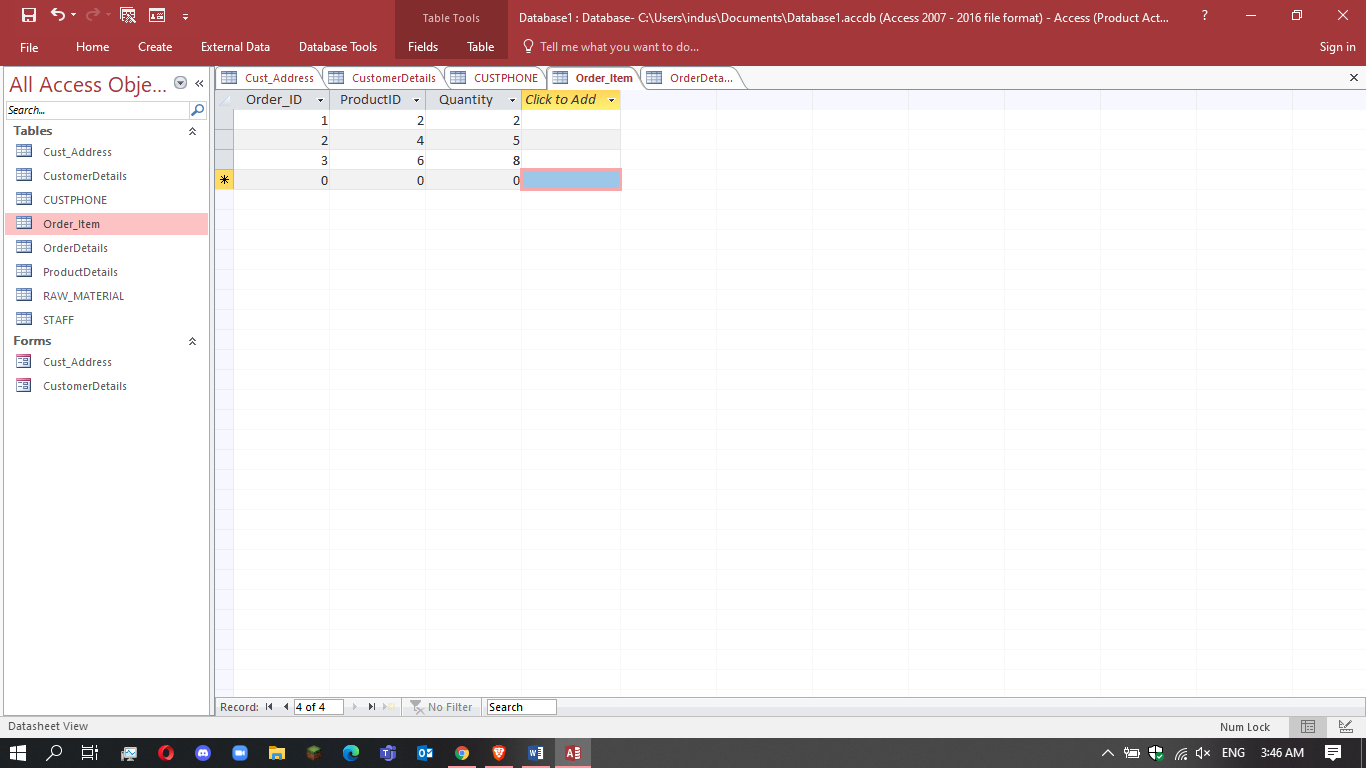
## 2nd Normal Form

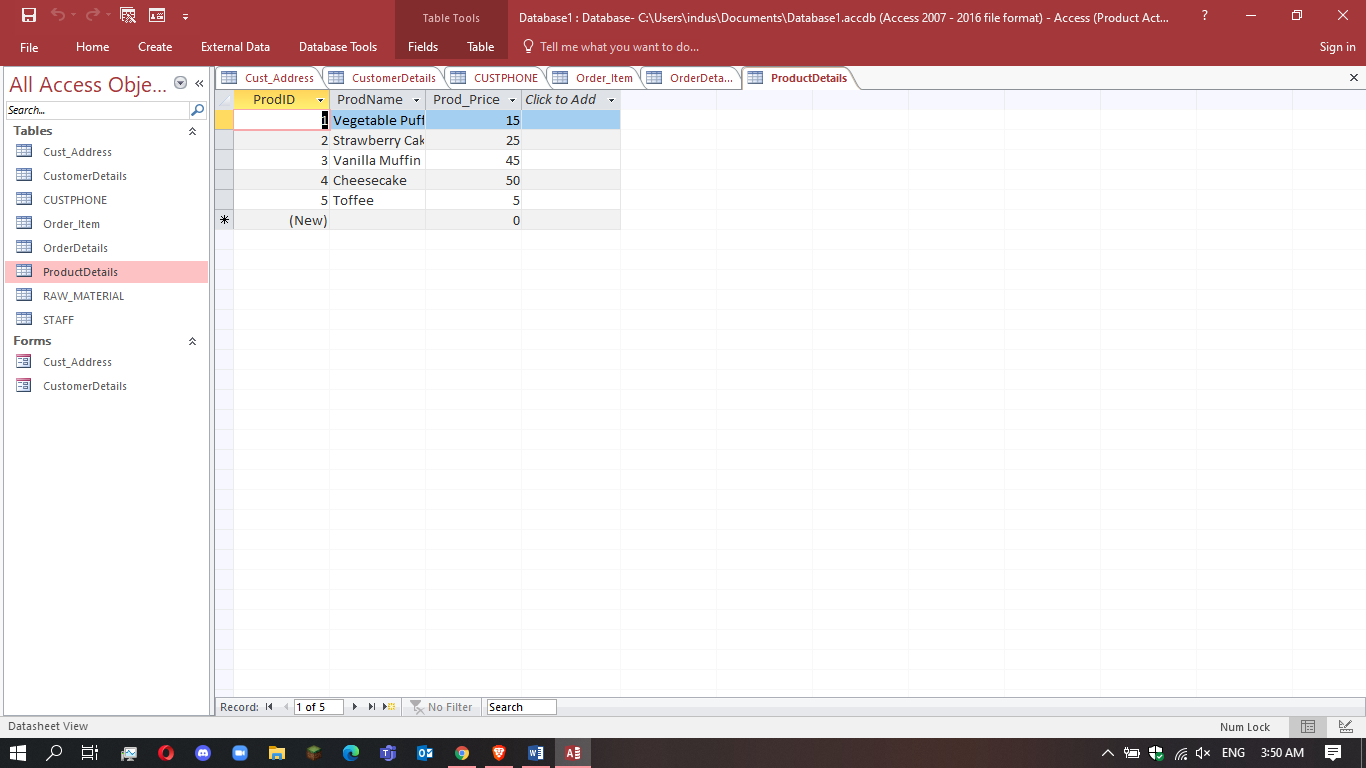
**Consider Order Table.**



Order table was not in 2NF as it had a partial dependency of Prod\_Name and Prod\_Price on Prod\_ID

So it decomposes into OrderDetails table and CustomerDetails table.

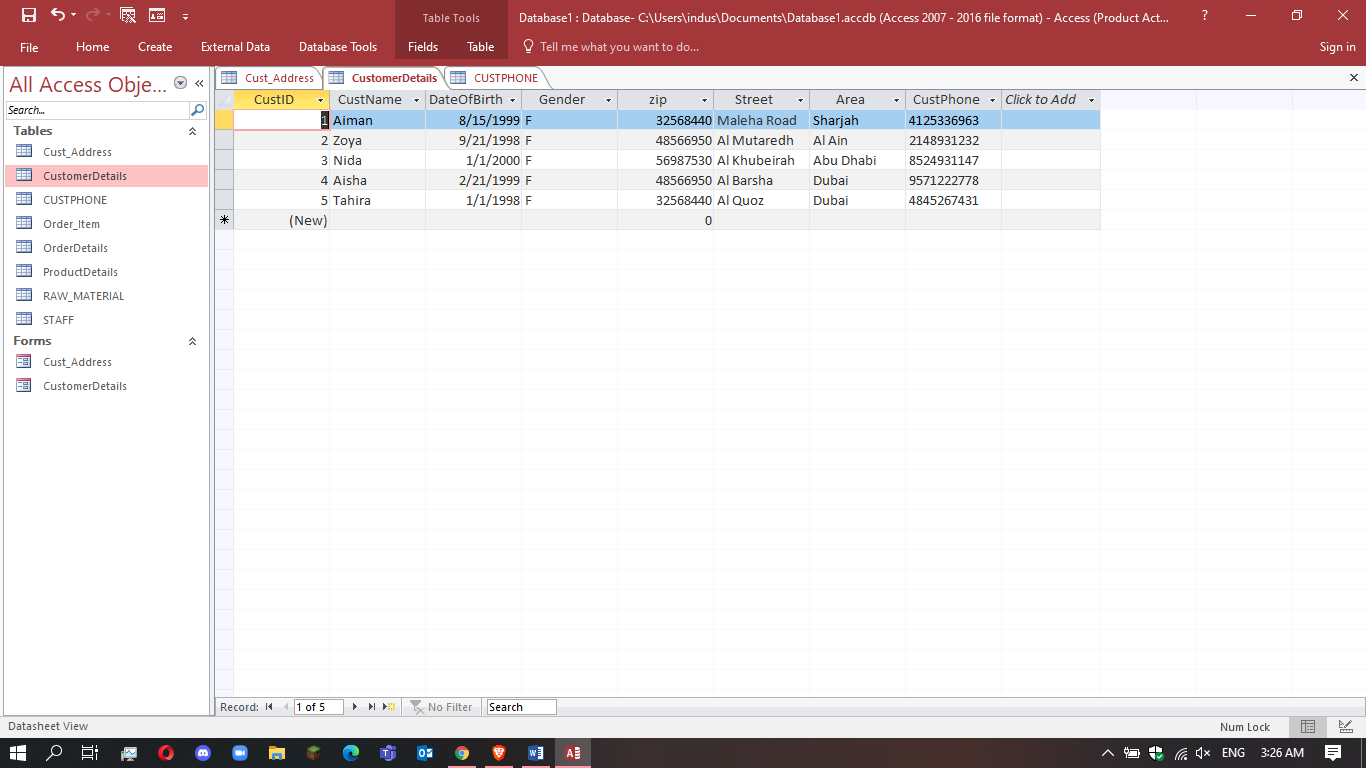




Now both tables are in 2NF.

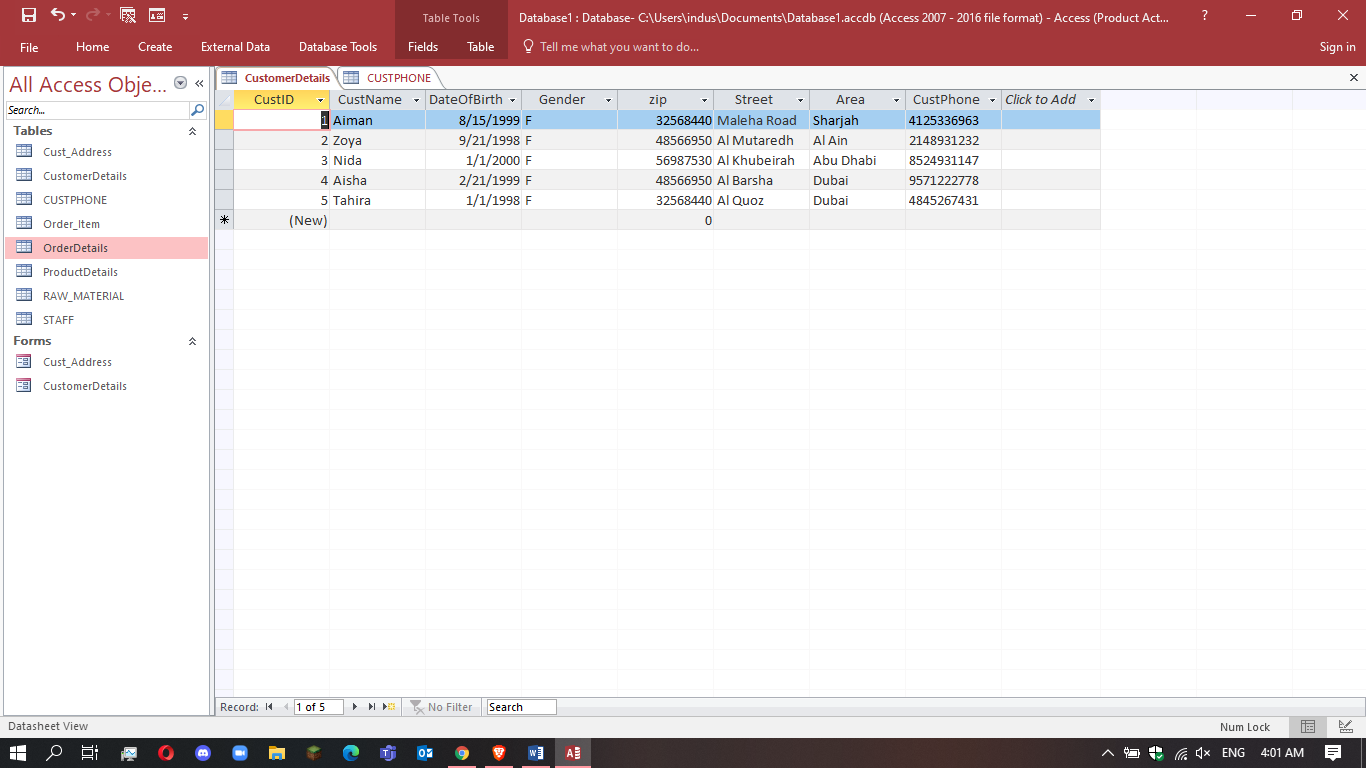
## 3rd Normal Form

**Consider Customer Table**



Customer table was not in 3NF as it had transitive dependency of street and area on zip,

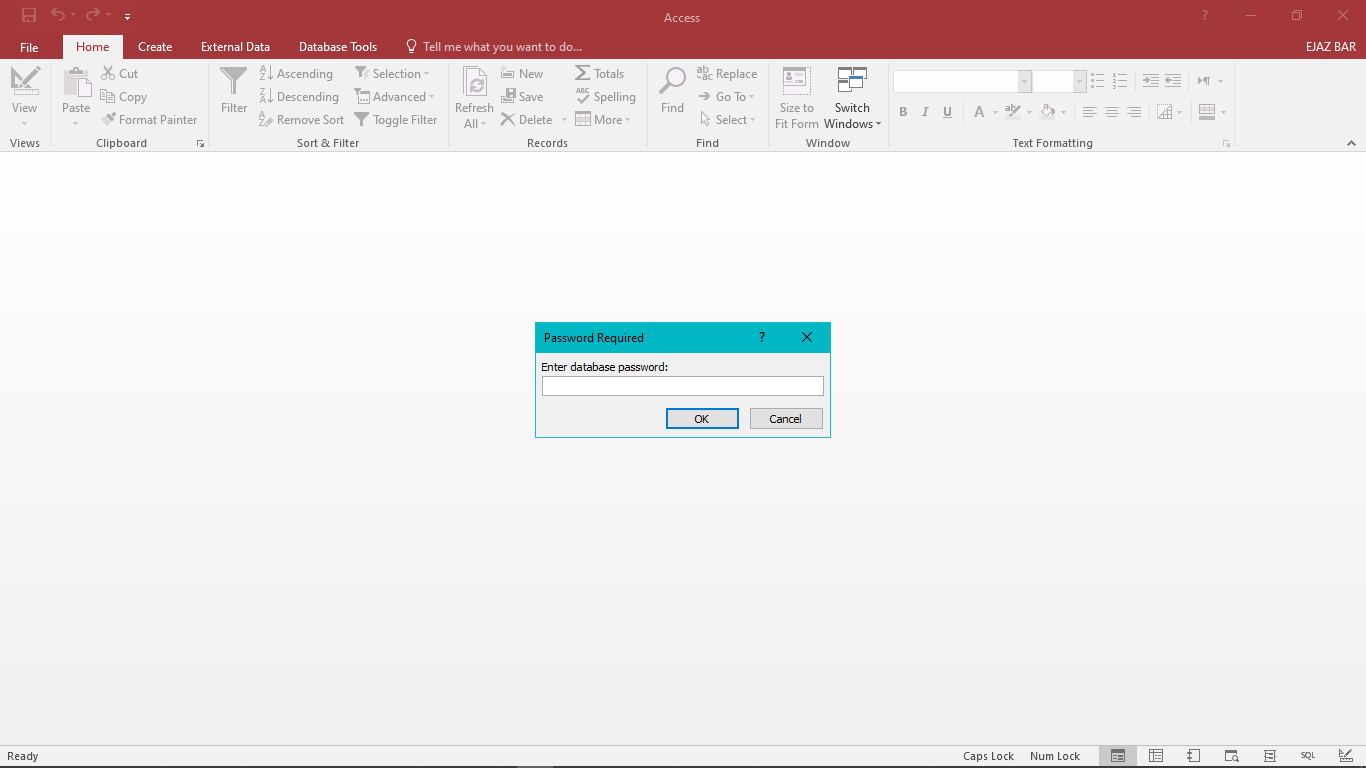
So it decomposes into customer details and CUSTPHONE.

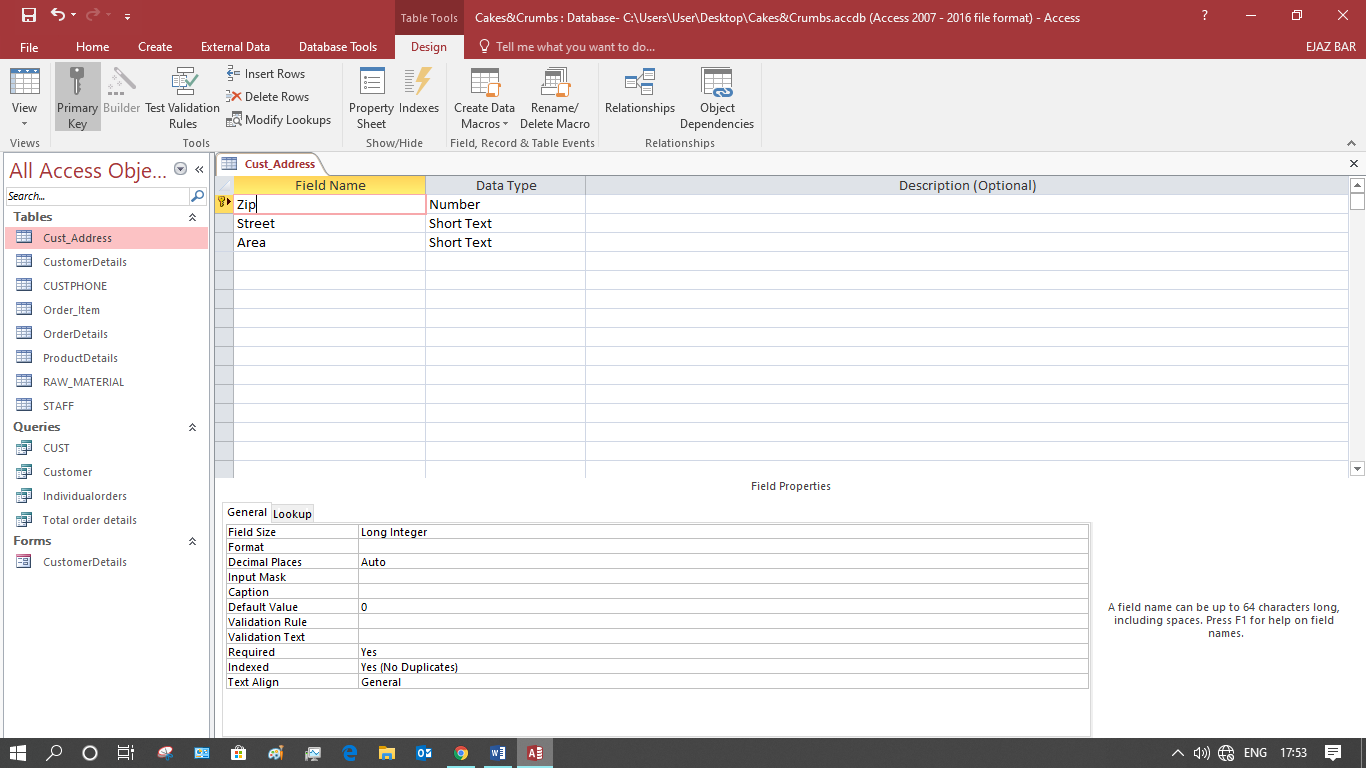


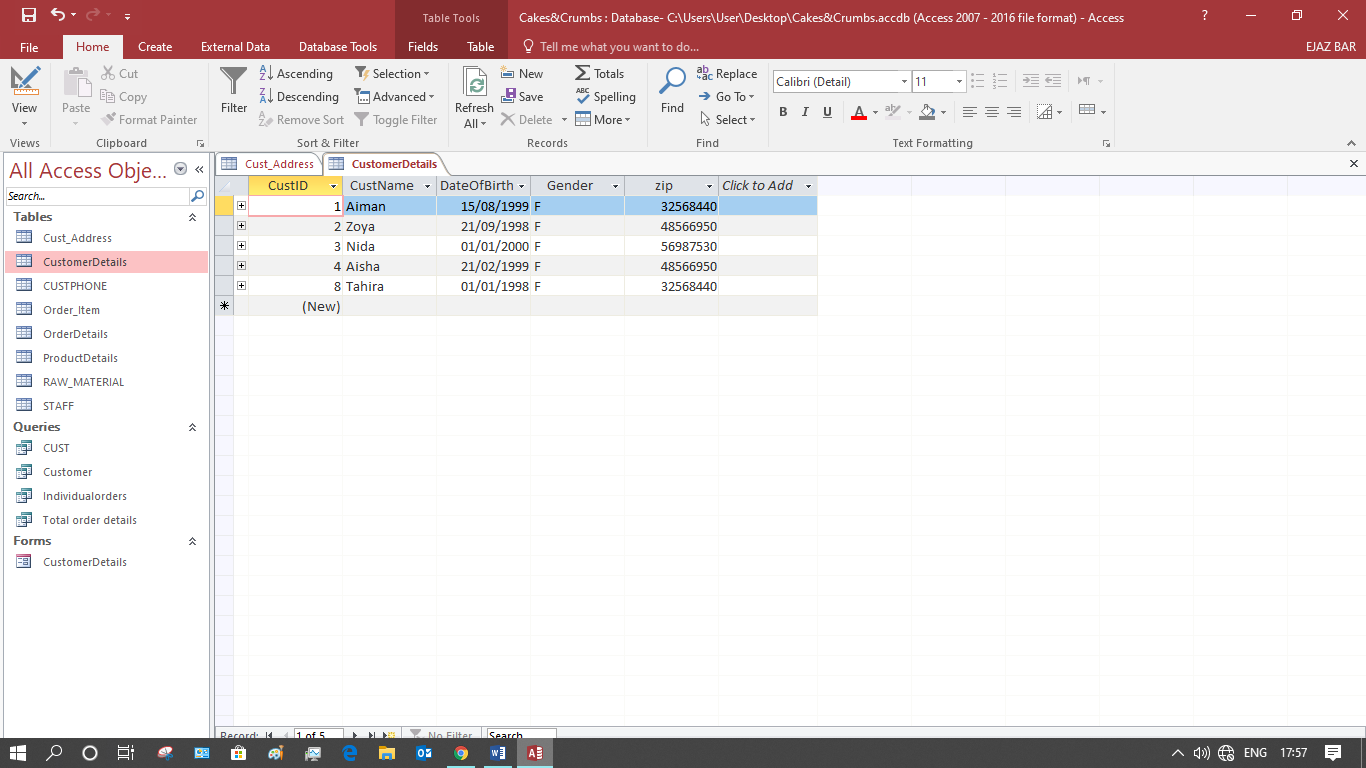
## 2.c) security and data integrity features

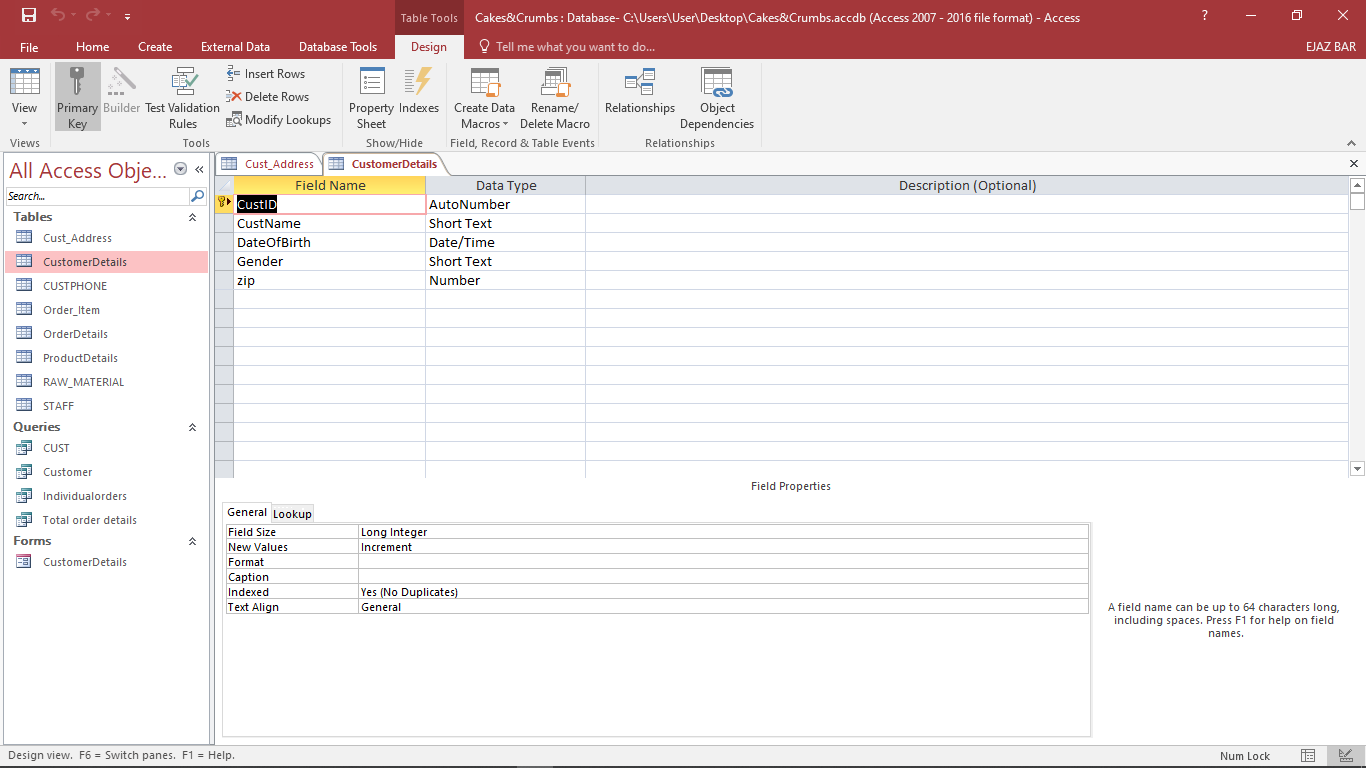
Data integrity

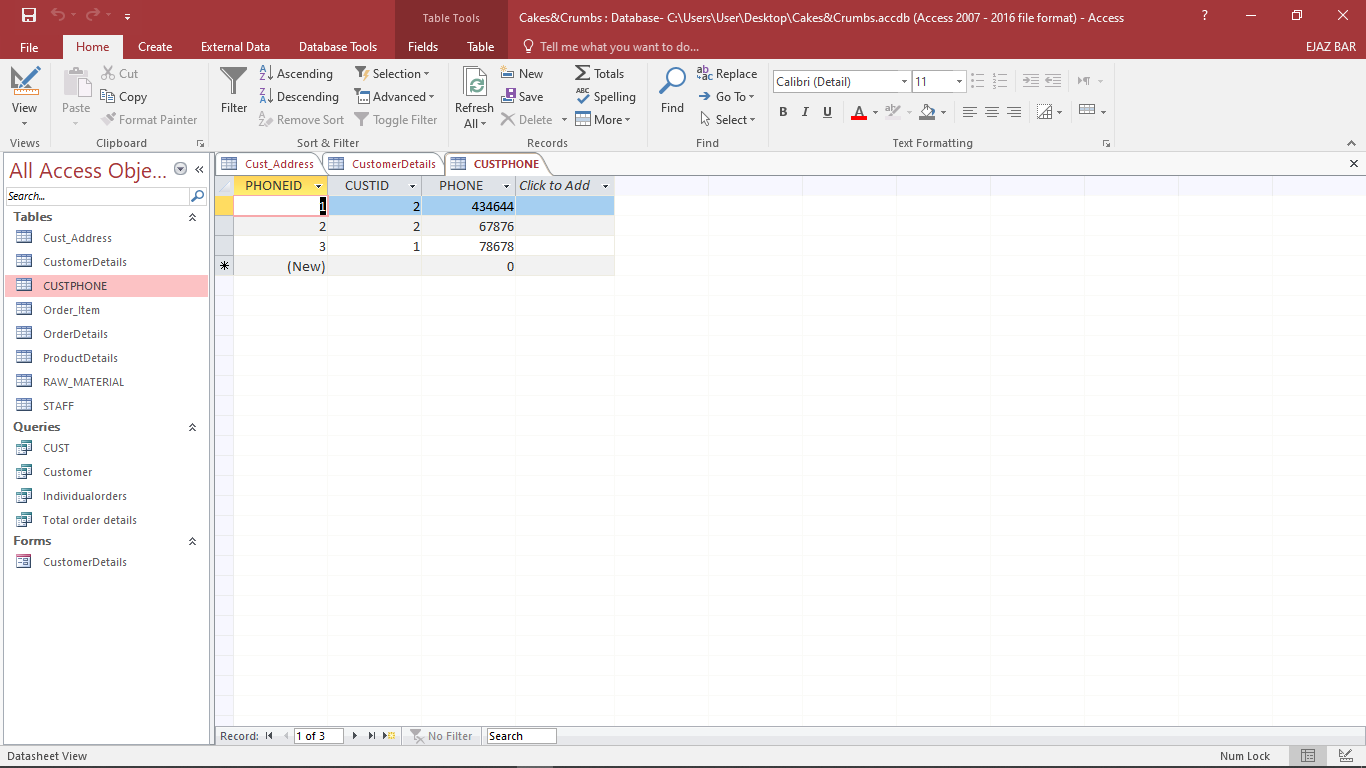
# THE DETAILS OF YOUR ENTITIES IN THE DATABASE

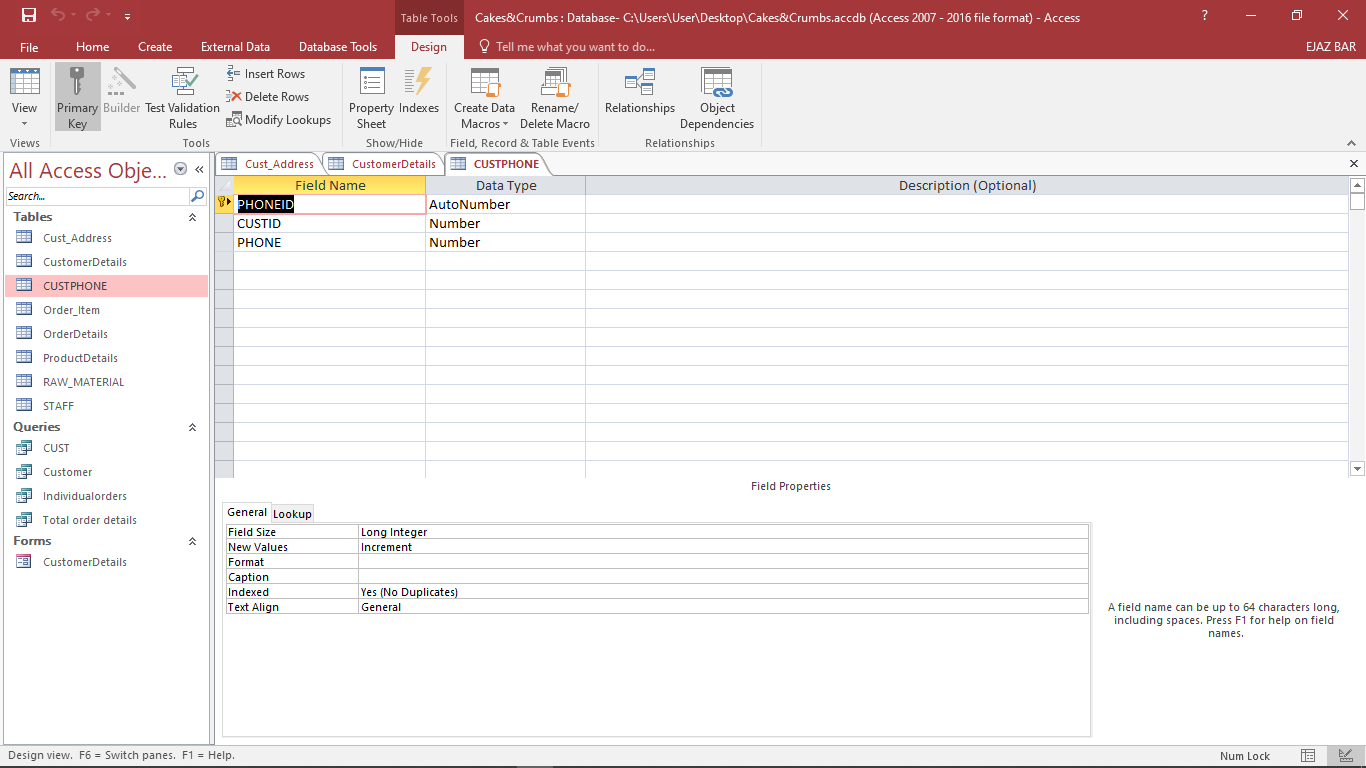


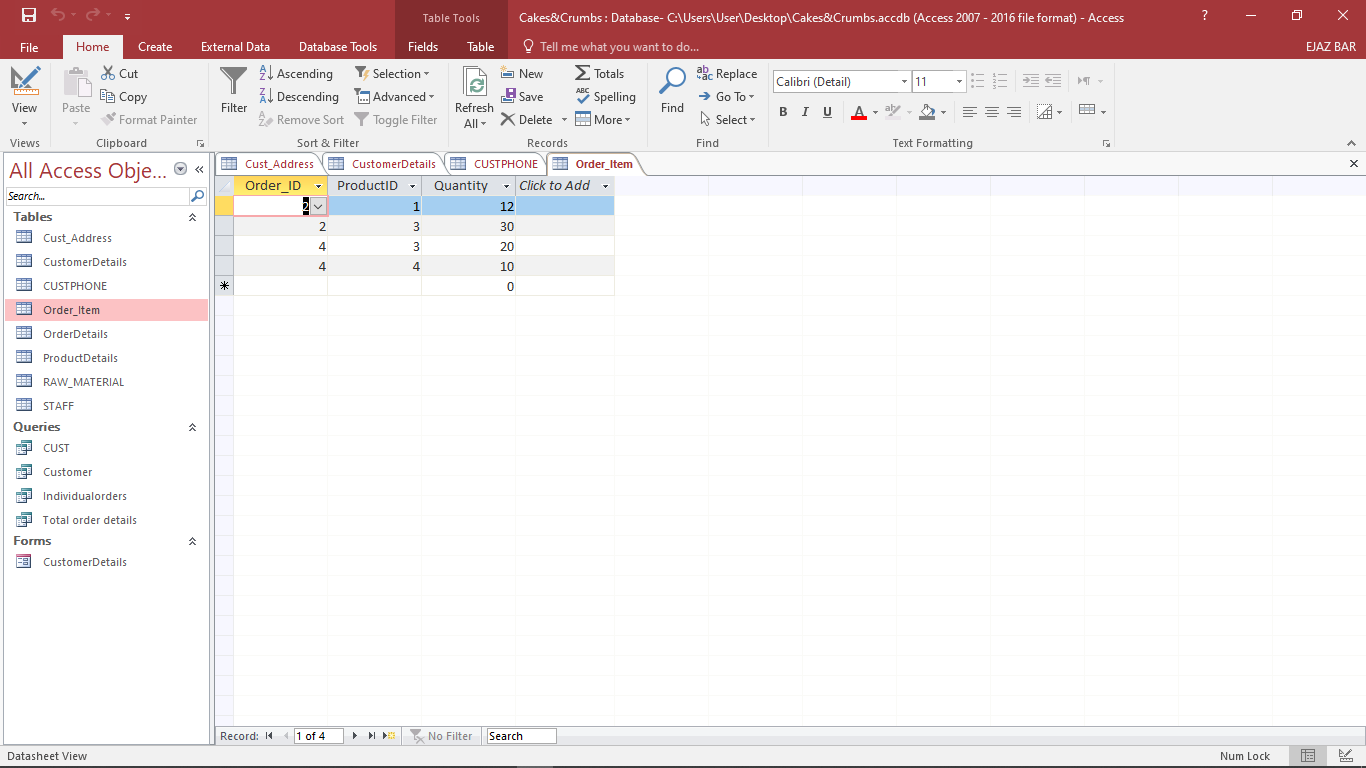


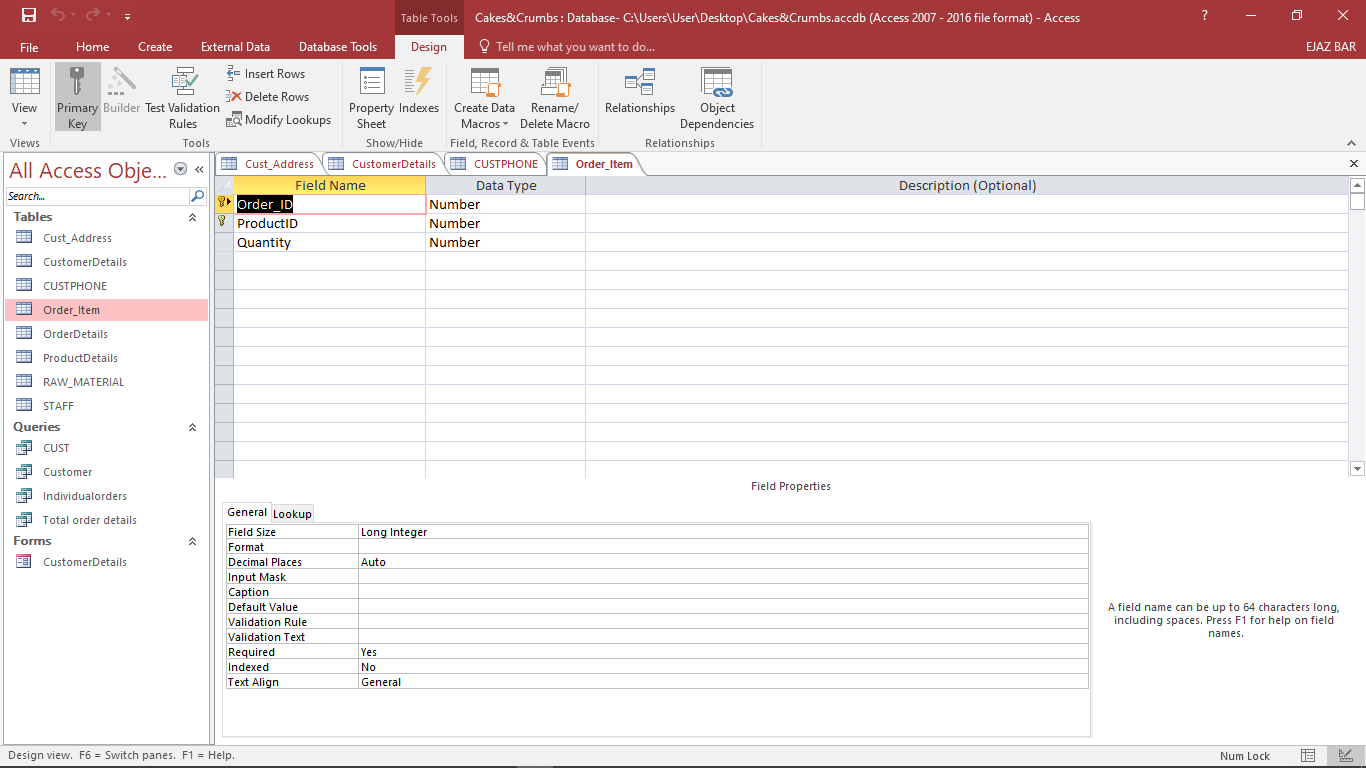


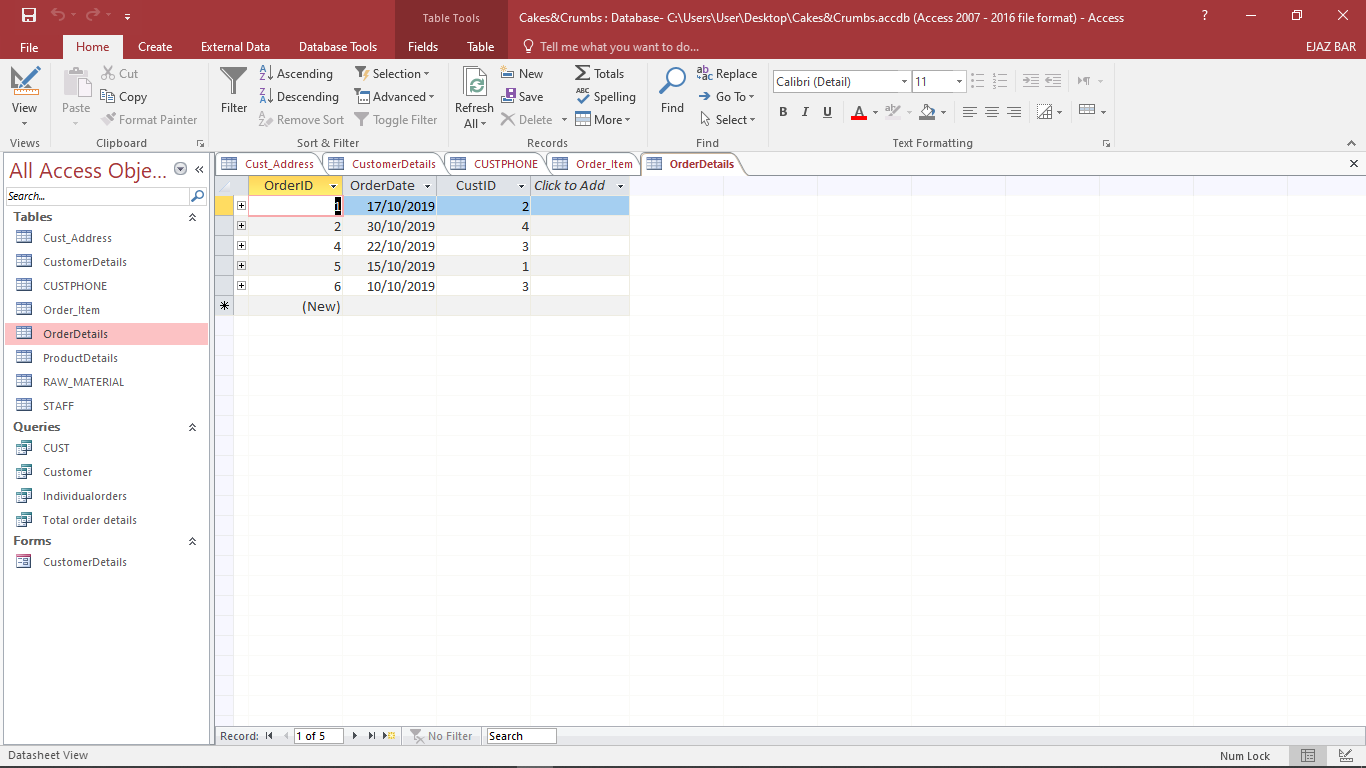


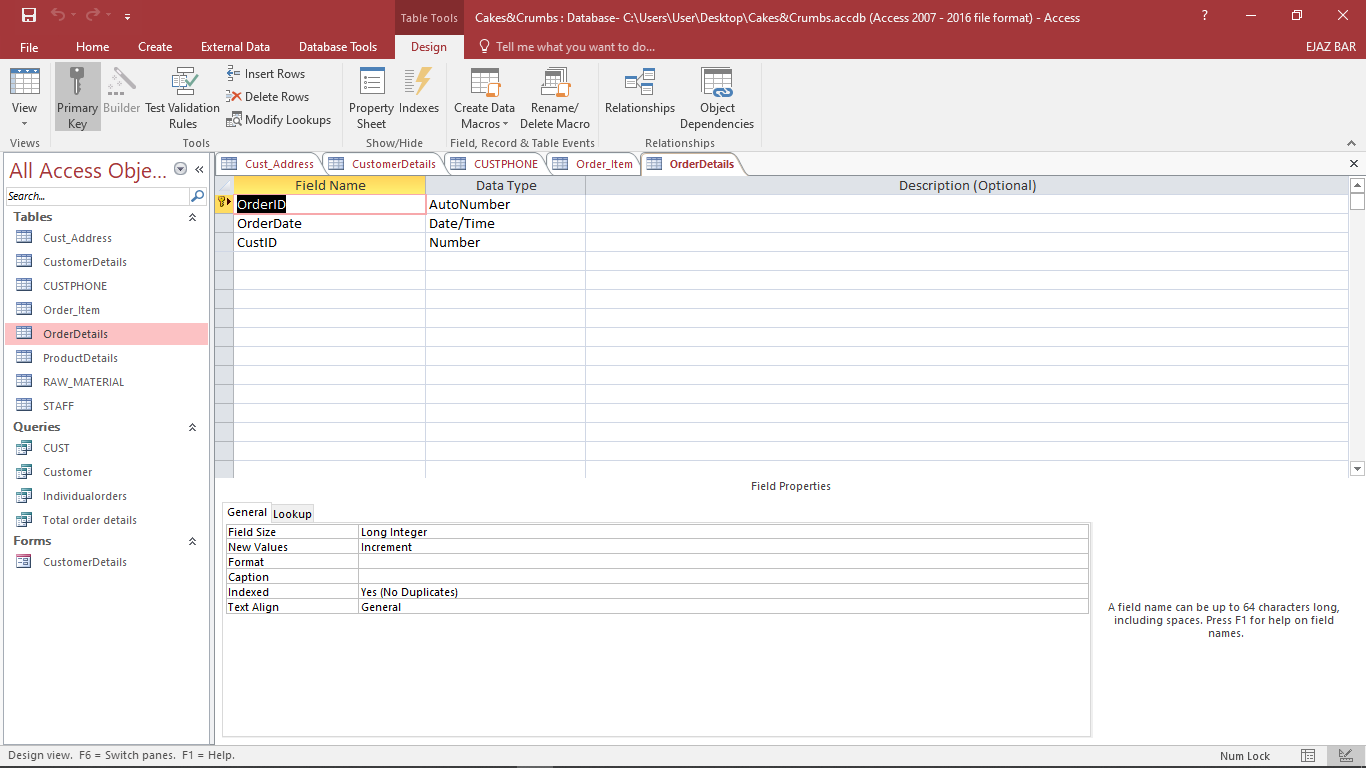


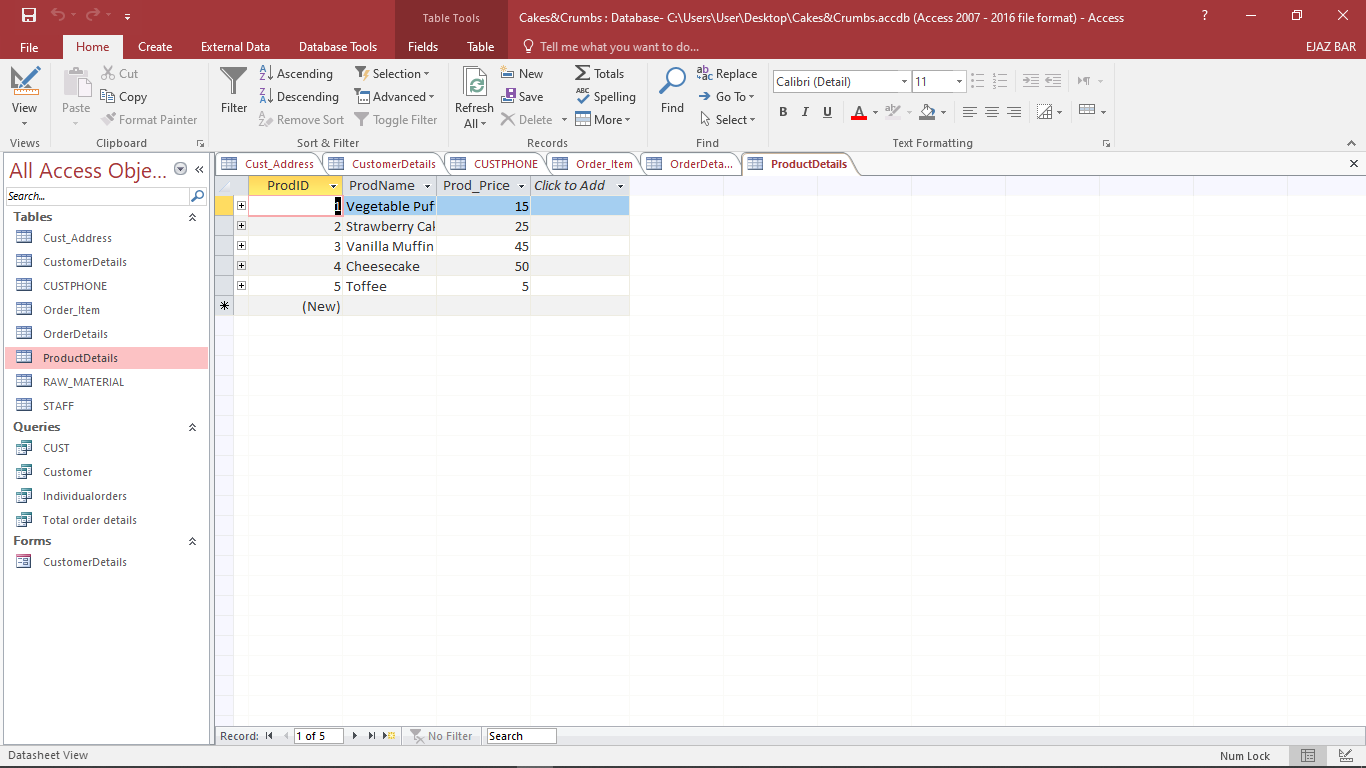


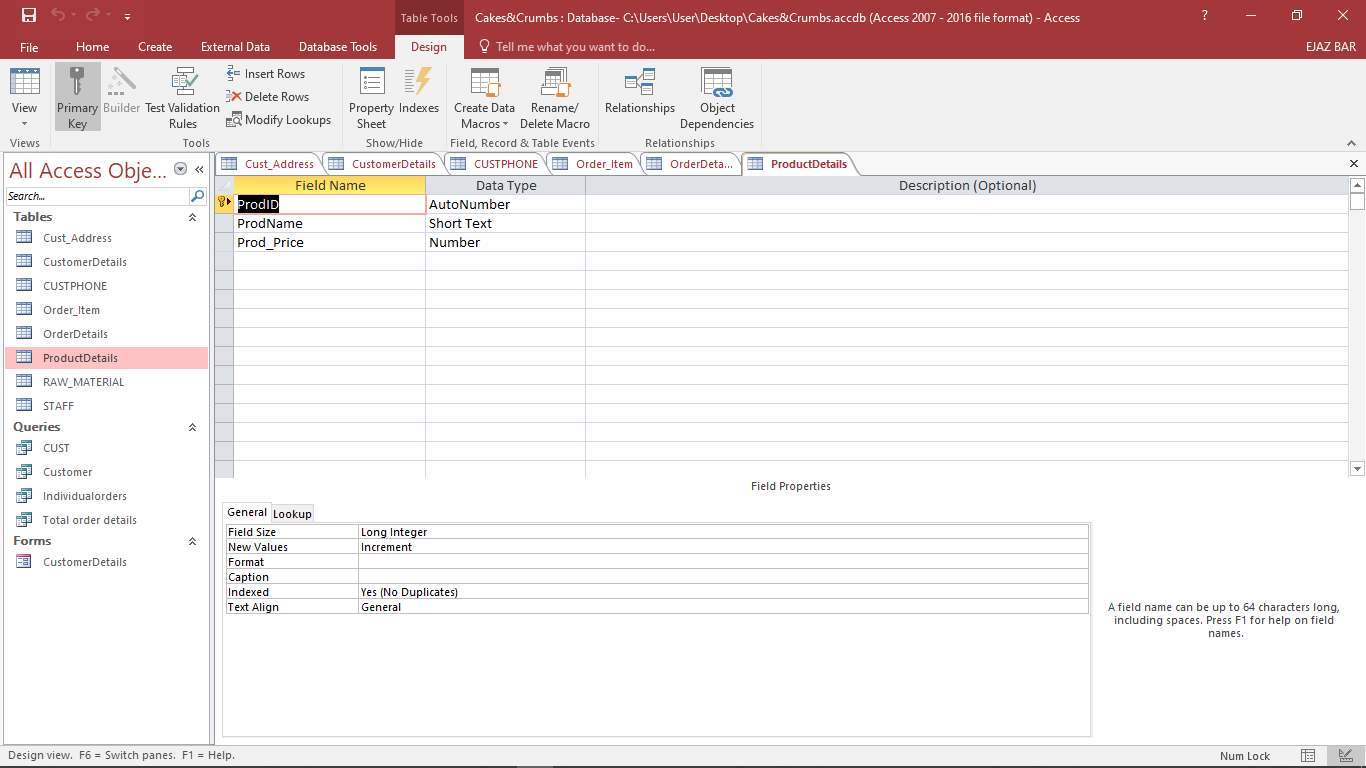


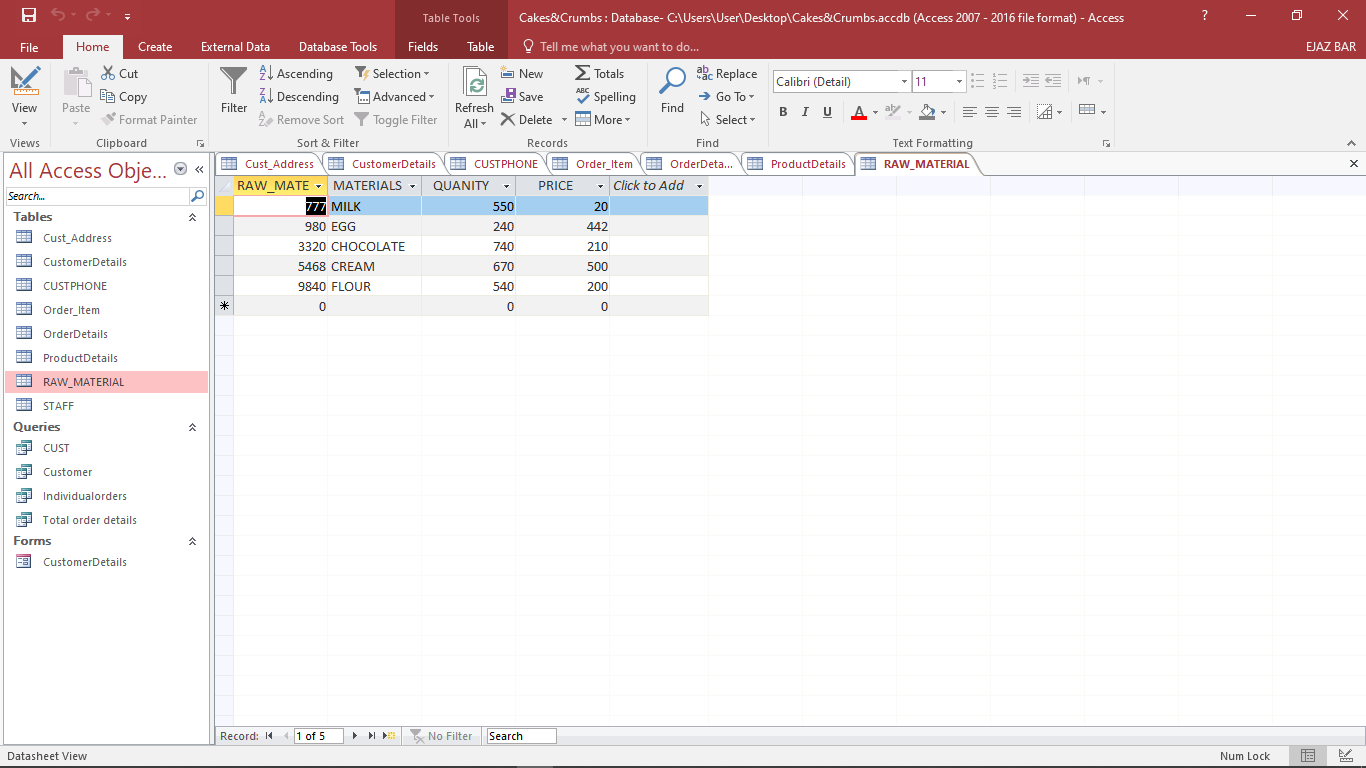


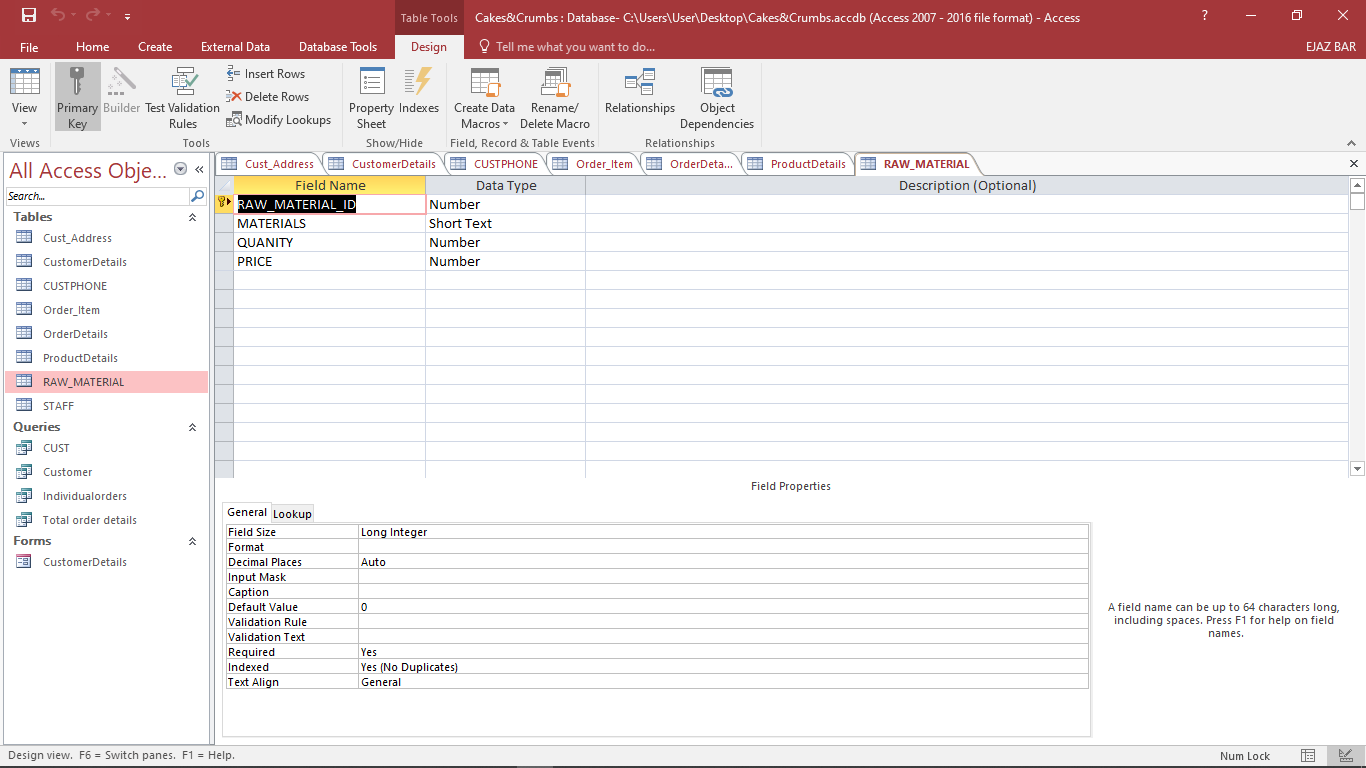


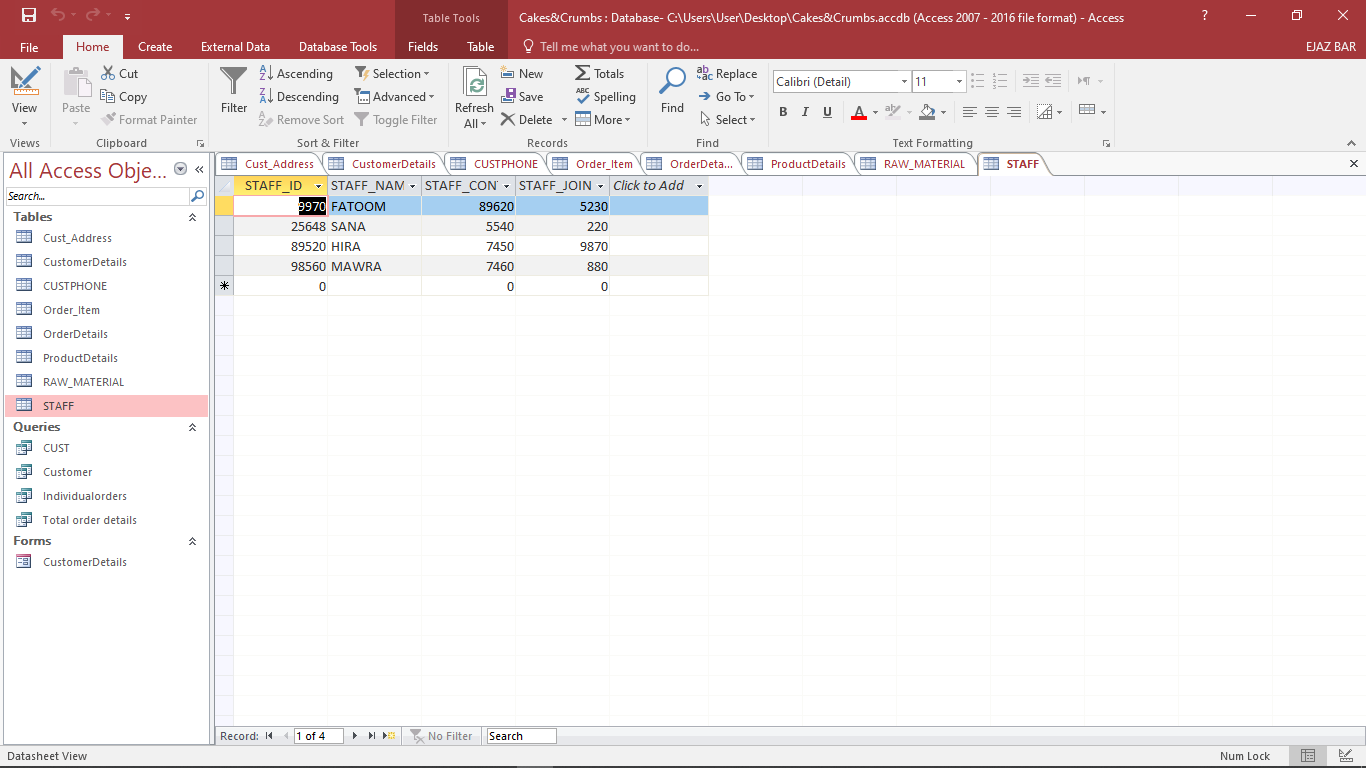


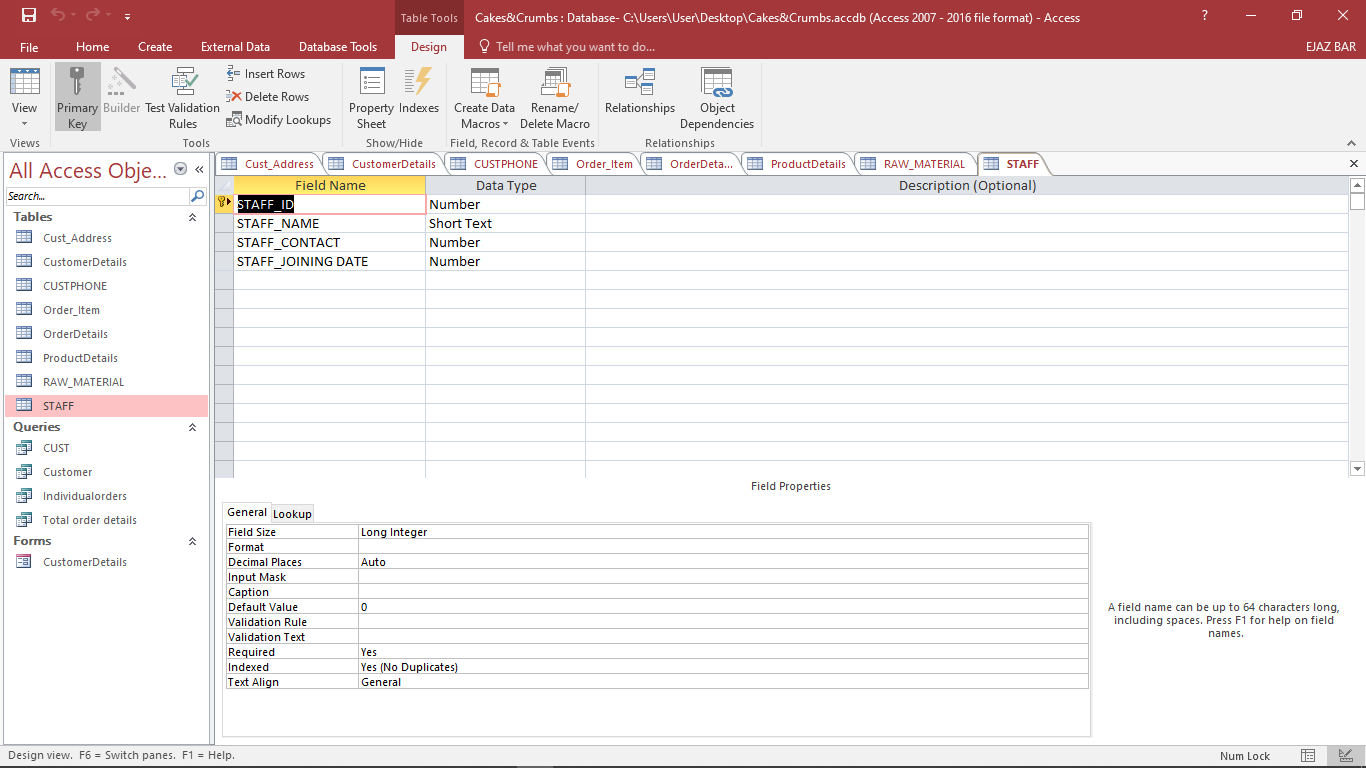












## Reference

Tutorialspoint, three level architecture of database. [Online] available at: <https://www.tutorialspoint.com/Three-Level-Architecture-of-Database> [accessed 21 July, 2021]

Guru99, Big data [online] available at: <https://www.guru99.com/what-is-big-data.html> [Accessed 21 July 2021].

Stefan Schmidt, 2013. the 3vs of big data [ online ] available at: <https://www.the-future-of-commerce.com/2013/03/07/the-3-vs-of-big-data/> [ Assessed 21 July 2021].

Java point, 2011. application of big data [online] available at: <https://www.javatpoint.com/applications-of-big-data> [Accessed 21 July 2021].

Funke, F. Kemper, A and Neumann, T., 2012. Compacting transactional data in hybrid. OLTP & OLAP databases. [Online] available at: <https://arxiv.org/abs/1208.0224> [accessed 21 august 2021].