Assignment 2:

Q1. Explain the difference(s) between data flow diagrams and flowcharts. Which do you think is easier to prepare? Why? Which do you think is easier to interpret? Why

- DFD represents the logical characteristics of the system, while flowcharts represent the physical ones. DFD helps to focus on information and it provides fixed documentation of data flows and processes, while flowcharts help to focus on technology. It reflects changes on physical components used in the information system. Although, both of them represent flow of data, but DFD ignores displaying units or department in which data are processed or stored. On the other hand, flowcharts displays data movements across offices or departments within a particular system.
- Flowchart is easier to prepare since DVD needs a higher skill to prepare it. DVD is more complex and that need to follow many constraints and rules to keep the size of DVD and its complexity manageable. In contrast, flowchart can be implemented by translating system description easily to symbols in a clear way, with no need to divide it into different levels of complexity.
- For interpretation, I think that if we have a simple system with few processes, it will be easier to interpret DFD since it has less symbols to learn. On the other hand, if we have more process which means more DFD levels and more complicity, then flowchart will be easier to interpret since it represents the whole system in only one level.

[1]: Dunn, C., Cherrington, J., & Hollander, A. (2005). Task Level Modeling. In *Enterprise information systems: A pattern-based approach* (3rd ed., pp. 102-113). Boston: McGraw-Hill/Irwin.

==========

Q2. Does every table in a relational database contain a foreign key? Explain.

- Foreign key is not necessary to be added in each table in the relational database. Foreign key is a primary key in an entity table but is posted as a foreign key into another table to represent a specific relationship between these two tables. So, sometimes we may have a table (department) with an ID (primary key) and its related attributes, and this ID can be posted as a foreign

key in other tables to represent different relationships, but department table may not contain foreign keys posted from other table.

[2]: Dunn, C., Cherrington, J., & Hollander, A. (2005). Task Level Modeling. In *Enterprise information systems: A pattern-based approach* (3rd ed., p. 123). Boston: McGraw-Hill/Irwin.

==========

Q3. Explain and illustrate the One Fact, One Place rule in relational database through your own example. Why is it important?

- One fact consists of a candidate key attribute value with another attribute value. This fact must appear only one time which means in only one place. It means we must ensure that no multiple facts in one place, and no one fact in multiple places. So, if we found any violation of the rule, we have to separate tables sometimes, we may create tables for other relationships, or we will need to poste primary keys to other tables as foreign key.[3] Here is an example:

Book_id	Book_name	Edtion	Author_name	BOD
1	The Nature	5th	Ahmed Jameel	1/10/1950
2	Life Around Us	2nd	Ali Khalid	5/12/1937
3	Masks	3rd	Faisal Ahmed	12/10/1925
4	The Dark	2nd	Ahmed Jameel	1/10/1950

In the above table we have one fact in multiple places, so we will need to separate this table into two tables and one for authors and another for book. We will need to add author_id as a primary key in author table, and then we will poste it as a foreign key on the book table to represent the relationship between them.

Author table:

Author_id	Author_name	BOD
1	Ahmed Jameel	1/10/1950
2	Ali Khalid	5/12/1937
3	Faisal Ahmed	12/10/1925

Book table:

Book_id	Book_name	Edtion	Author_id
1	The Nature	5th	1
2	Life Around Us	2nd	2
3	Masks	3rd	3
4	The Dark	2nd	1

So, using this rule is so important to avoid any redundancy of data and repeating group which may affect your database performance. Also, it keeps your relational database consistent all the time.

[3]: Dunn, C., Cherrington, J., & Hollander, A. (2005). Relational Database Design: Converting Conceptual REA Models to Relational Database. In *Enterprise information systems: A pattern-based approach* (3rd ed., p. 124). Boston: McGraw-Hill/Irwin.

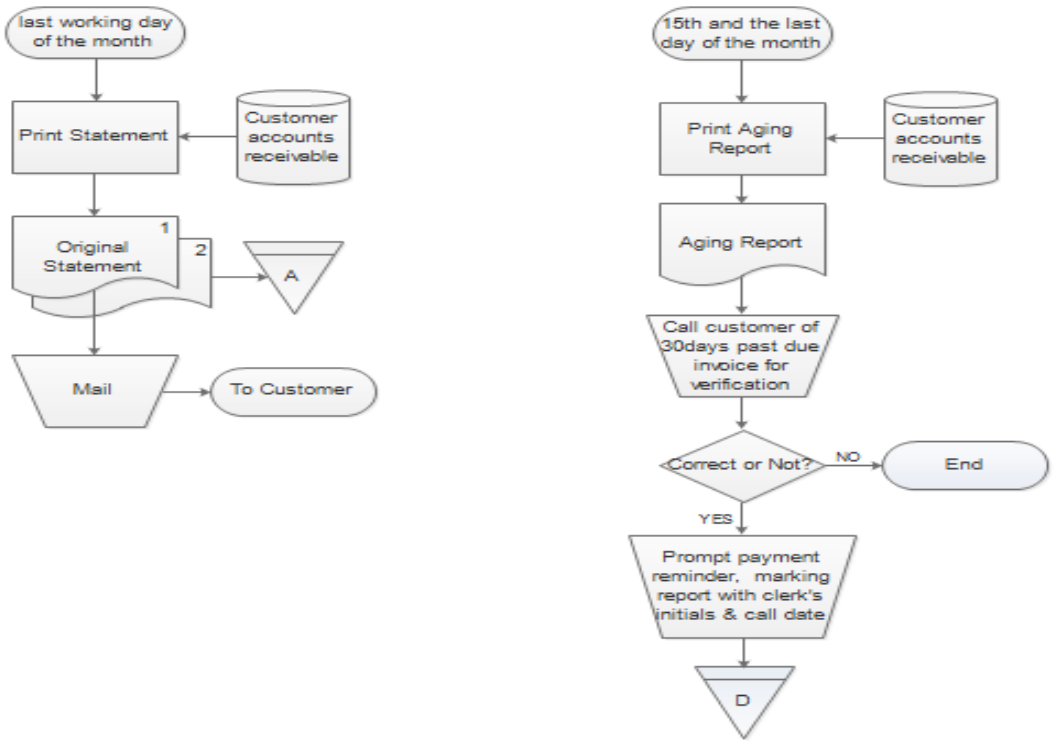
==========

Q4. Case Study of “Convenient Computing Associates”

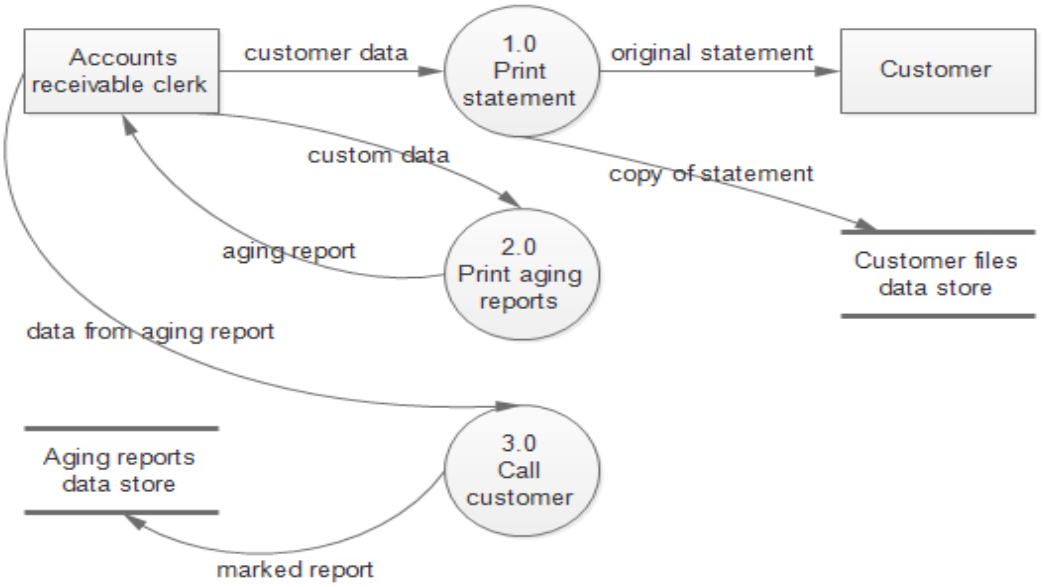
Convenient Computing Associates (CCA) operates a mailing-order operation and sells a vast array of computer products and accessories. The following is a description of Convenient Computing's collection process: The policy of CCA is to collect all accounts receivable as quickly as possible. CCA encourages prompt payment by sending reminder statements at the end of each month and calling customers whose balance is more than 30 days past due. On the last working day of the month, the accounts receivable clerk prints a statement for each customer showing a balance due on the account. Statements are automatically generated by a computer which maintains the accounts receivable files and records. The original statement is mailed to the customer and a copy of the statement is filed alphabetically according to customers’ last names. On the 15th and the last day of the month, the accounts receivable clerk prints an open accounts receivable aging report by customer. The clerk calls the customer on all invoices more than 30 days past due. The purpose of the call is to verify the accuracy of the invoice data. If the information is correct, a polite reminder is given to encourage prompt payment. The open receivable report is marked with the clerk's initials and the date the call was made, and filed by date when all calls are complete. Required:

a. Prepare a flowchart of this process.
b. Prepare a logical data flow diagram for this process.

- Flowchart for this process:

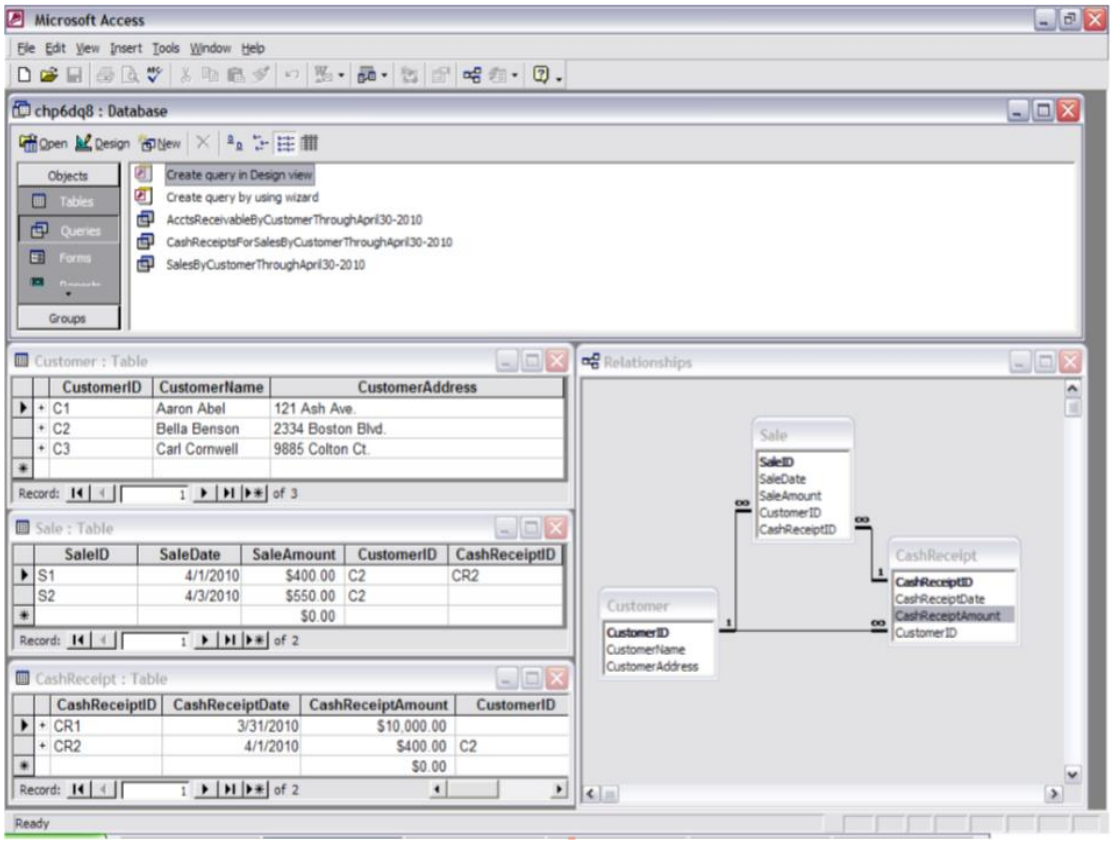


- Logical data flow diagram for this process:

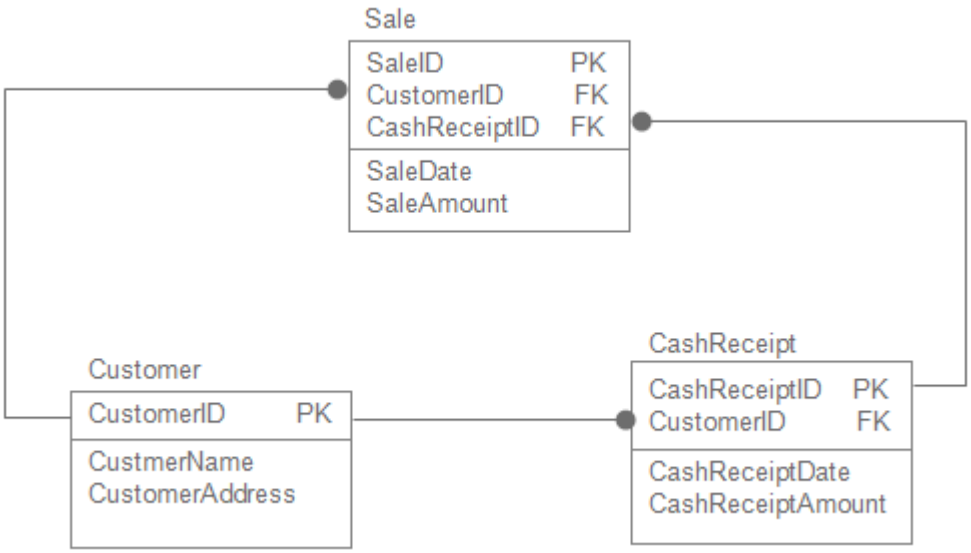


==========

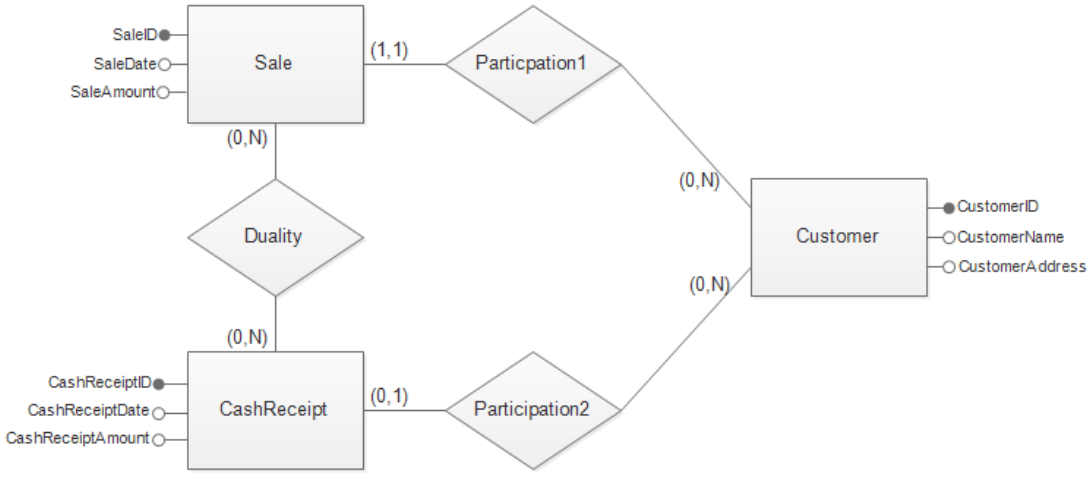
Q5. Inspect the physical database components implemented in Microsoft Access as displayed below. Construct the likely underlying logical and conceptual models.



- Logical model:



- Conceptual model:



==========