Information and Database Management Systems I

(CIS 4301 UF Online)

Fall 2024

Instructor:Alexander Webber

TA: Kyuseo Park

Homework 1

Printed Name:	
UFID:	
Email Address:	

Instructions: Please provide your answers to the questions of the following pages in Word or handwritten on separate sheets of paper. Mark clearly to which question each answer belongs. Then convert or scan your work into PDF (the latter by using either a scanner or a suitable scanner app on your smartphone). Note that *only the PDF format* is allowed and that your submission must be a *single PDF file*. Finally, upload your PDF file into *Canvas* and follow the instructions there.

Note: All homework assignments are designed for a period of two, three, or even four weeks (see course deadline sheet). This means they cannot be solved in two or three hours but require a considerable amount of time and effort. Therefore, the first recommendation is to start with them as soon as they are posted. The second recommendation is to distribute the work on a homework assignment over the entire available period. The third recommendation is to submit the homework solutions *on time before the deadline*.

Pledge (Must be signed[1] according to the UF Honor Code):

On my honor, I have neither given nor received unauthorized aid in doing this assignment.

_____

Student signature

[1] Each student is obliged to print out this page, fill in the requested information in a handwritten and readable manner, make the *handwritten* signature, scan this page into PDF, and put this page as the first page of the PDF submission.

Question 1 (ER Model – ER Diagram Design (I)) [24 points]

Design two equivalent Entity-Relationship diagrams, one in Chen's notation and one in crow's foot notation, that model a *hospital management system* and consider the requirements listed below. That means that you have to identify and determine suitable entity sets, relationship sets, attributes, keys of entity sets (if not specified), and so on. Further, add the cardinalities to the relationship sets and write down your assumptions regarding the cardinalities if there could be a doubt.

The requirements are:

- Each patient has a unique ID and is described by name, address, age, phone number, and date of birth. Age is a derived attribute. A patient's address includes street, city, state, and zip code.
- At least one medical record is recorded for each patient.
- Doctors have their specialties, date of birth, age, and address, and treat multiple patients. Age is a derived attribute. Their unique identifier is their name. A doctor's address includes street, city, state, and zip code.
- Nurses are also identified by their names, have specialties, and share personal details with doctors. They provide care to various patients.
- Medical records, which are distinctively identified by a record ID, contain treatment details.
- Appointments have a unique ID, encompass date and time, and involve specific patients and doctors.
- A patient can have multiple appointments.
- While a doctor can attend multiple appointments, each appointment is managed by only one doctor.
- Nurses and doctors belong to a hospital department.
- Hospital departments have by a unique ID, names, functions, and a department head.
- Each prescription is linked to a specific medical record. Prescriptions are detailed with medication and dosage and have a unique ID.

NOTE: Use ERDPlus to draw both ER diagrams. This will help you become prepared for Exam 1 where we will use ERDPlus. ERDPlus allows one to export an ER diagram as an image file (with the file ending `.png`) or as a text file in an ERDPlus specific ASCII format (with the file ending `.erdplus`). Create an account on the ERDPlus web page to be able to store ER diagrams so that you can continue or modify them later.

The main graphical notation supported in ERDPlus is the *crow's foot* notation that is a successor of Chen's notation and has a number of syntactical (but not semantic) differences. The textbooks as well as the instructor's special session about ERDPlus will help you understand and distinguish the two different notations. ERDPlus also supports Chen's notation by using straight edges and *labels* attached to the edges. But a first problem is that due to a conceptual error (or an implementation error) in ERDPlus, text boxes containing labels cannot be minimized with the mouse in a manner so that they closely surround the label. Their minimal size is relatively large so that they can superimpose other components of an ER diagram. A way to solve this problem is to export your ER diagram into the ERDPlus specific ASCII format (with the file ending `.erdplus`), use a text editor to change the height and width of each label that overlaps other components of the ER diagram, delete the old file or rename the old file of the ER diagram in ERDPlus, and import the modified file into ERDPlus. For labels such as m, n, or 1, the value 15 for height and width is suitable. More precisely, for such labels, the string `"width":50,"height":50` in the text file should be replaced repeatedly by the string `"width":15,"height":15`. A second problem is that ERDPlus is unable to generate double lines for relationship sets since they are not needed in the crow's foot notation but used in Chen's notation to indicate total participation. The only chance to achieve this is to use an image program (such as Paint 3D on Windows) to edit the exported image file of the ER diagram and add the missing second line where needed. The solutions to both problems are inconvenient but acceptable.

Question 2 (ER Model – ER Diagram Design (II)) [24 points]

Design two equivalent Entity-Relationship diagrams, one in Chen's notation and one in crow's foot notation, that model a *shared bicycle system* and consider the requirements listed below. That means that you have to identify and determine suitable entity sets, relationship sets, attributes, keys of entity sets (if not specified), and so on. Further, add the cardinalities to the relationship sets and write down your assumptions regarding the cardinalities if there could be a doubt.

The requirements are:

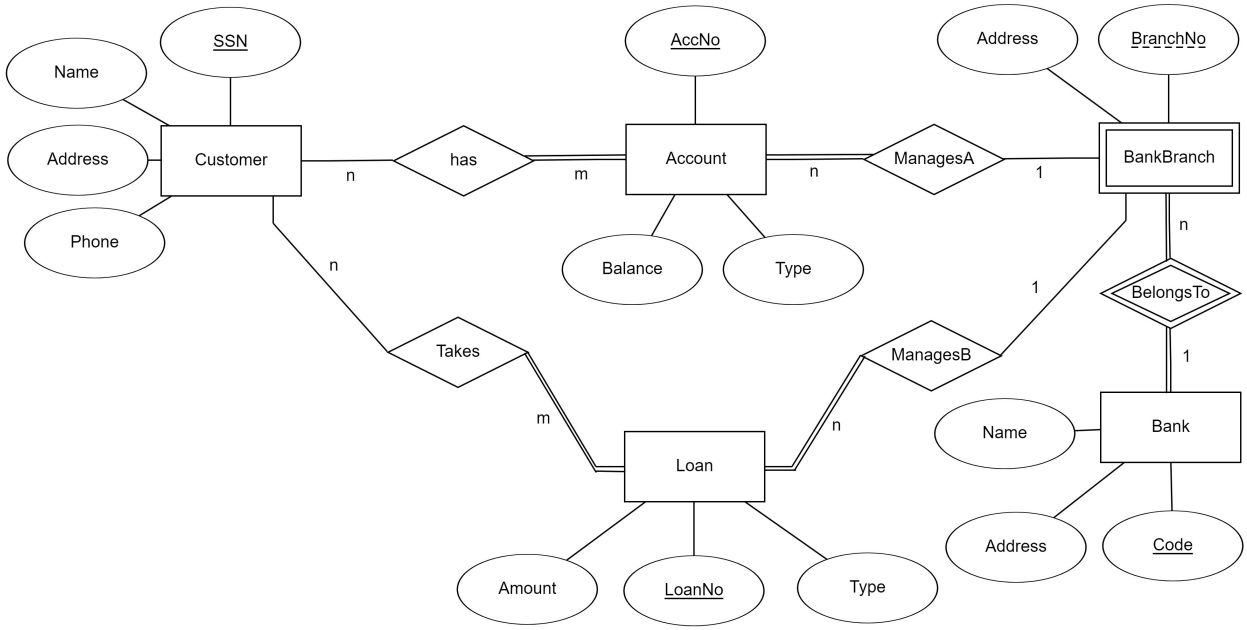
- Each bicycle in the system is identified by a unique bicycle ID and includes details such as its type, last maintenance date, and status.
- Users of the system are identified by a unique user ID and include personal information such as name, payment information, and phone number.
- Stations where bicycles are docked are identified by a unique station ID and include information such as location, capacity, and the number of available bikes.
- A user can rent multiple bicycles over time.
- Bicycle rentals are tracked with a unique rental ID and include details such as start and end times, and the stations where the rental starts and ends.
- Payments made by users are identified by a unique payment ID and contain details such as the amount, date, and payment method.
- Bicycle inspections are recorded with a unique inspection ID and include the date of an inspection, inspector's details, inspection results, and notes.
- A bicycle can be docked at one station at a time.
- A user makes several payments for different rentals.
- Each bicycle undergoes periodic inspections.
- Users can provide feedback for their bicycles. A feedback can have a comment and a date.

NOTE: Use ERDPlus to draw both ER diagrams. This will help you become prepared for Exam 1 where we will use ERDPlus. ERDPlus allows one to export an ER diagram as an image file (with the file ending `.png`) or as a text file in an ERDPlus specific ASCII format (with the file ending `.erdplus`). Create an account on the ERDPlus web page to be able to store ER diagrams so that you can continue or modify them later.

The main graphical notation supported in ERDPlus is the *crow's foot* notation that is a successor of Chen's notation and has a number of syntactical (but not semantic) differences. The textbooks as well as the instructor's special session about ERDPlus will help you understand and distinguish the two different notations. ERDPlus also supports Chen's notation by using straight edges and *labels* attached to the edges. But a first problem is that due to a conceptual error (or an implementation error) in ERDPlus, text boxes containing labels cannot be minimized with the mouse in a manner so that they closely surround the label. Their minimal size is relatively large so that they can superimpose other components of an ER diagram. A way to solve this problem is to export your ER diagram into the ERDPlus specific ASCII format (with the file ending `.erdplus`), use a text editor to change the height and width of each label that overlaps other components of the ER diagram, delete the old file or rename the old file of the ER diagram in ERDPlus, and import the modified file into ERDPlus. For labels such as *m*, *n*, or 1, the value 15 for height and width is suitable. More precisely, for such labels, the string `"width":50,"height":50` in the text file should be replaced repeatedly by the string `"width":15,"height":15`. A second problem is that ERDPlus is unable to generate double lines for relationship sets since they are not needed in the crow's foot notation but used in Chen's notation to indicate total participation. The only chance to achieve this is to use an image program (such as Paint 3D on Windows) to edit the exported image file of the ER diagram and add the missing second line where needed. The solutions to both problems are inconvenient but acceptable.

Question 3 (ER Model – Chen's Notation versus (Min, Max)-Notation) [20 points]

Let us assume the following ER diagram in Chen's notation that describes a banking scenario.



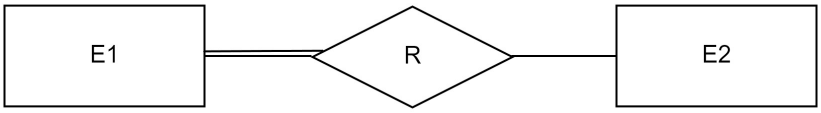
Answer the following questions. Provide detailed explanations.

(a) [2 points] Why is *BankBranch* modeled as a weak entity set?

(b) [4 points] Explain the sub-scenario *Bank – BelongsTo – BankBranch*. Why is the weak entity set *BankBranch* connected to the relationship set *BelongsTo* by a double line? Is this double line optional, that is, could it also be a single line? Why is the entity set *Bank* connected to the relationship set *BelongsTo* by a double line? Is this double line optional, that is, could it also be a single line?

(c) [10 points] List the names of all relationships sets and specify the (*min*, *max*) constraint on each participation of an entity set in a relationship set. Explain and justify your choices in full detail.

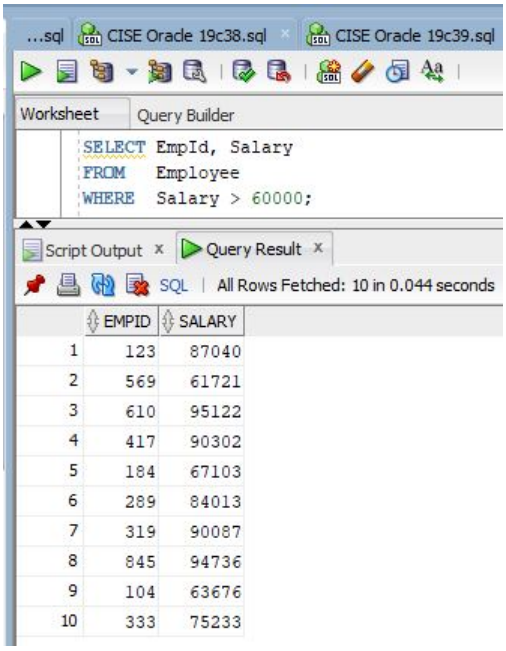
(d) [4 points] Consider the following incomplete ER diagram in Chen's notation.



Attributes are ignored. The cardinalities on the edges are missing. Draw all ER diagrams with all possible cardinalities in (*min*, *max*)-notation that can be derived from this ER diagram. What is the name of each such relationship set in the ER model (example: one-to-one)?

Question 4 (Oracle – First Steps in SQL) [32 points]

The objective of this exercise is to use your CISE Oracle account to design a table schema, create the (empty) table, fill the table with tuples, and answer a number of queries on this table. Provide **SQL statements** for all tasks and colloquial queries. Show your SQL commands, queries, *and* the outputs of all results as **screenshots** in **Oracle SQL Developer**. The following figure shows an example how such a screenshot should look like:



On Windows, for example, the *Snipping Tool*, *Lightshot*, and *Greenshot* are nice tools to create screenshots. Corresponding tools for Linux are, for example, *Spectacle*, *Shutter*, and *ScreenCloud*. MacOS provides similar tools that you should look up yourself.

Some needed SQL commands not taught in class so far

In some SQL statements, you will have to use the logical connectives AND and OR to connect subconditions: <condition1> AND <condition2>, <condition1> OR <condition2>. Example:

```
SELECT UFID
FROM   Student
WHERE  Name = 'Smith' AND Sem > 3;
```

The ORDER BY statement followed by a comma separated list of attribute names allows one to order tuples. This statement may only appear *at the end* of a SQL statement. Sorting is performed for each attribute separately and consecutively from right to left in the attribute list[1]. This means that attributes listed first have higher order priority than attributes following them. Adding the keywords ASC or DESC behind each attribute means that the values of the corresponding attributes are sorted in *ascending* or *descending* order respectively. Example:

```
SELECT   *
FROM     Student
ORDER BY Name ASC, Address DESC;
```

[1] Compare this to multi-attribute sorting (level sorting) in the spreadsheet program Excel. The underlying sorting concepts are the same in both tools.

This statement should be distinguished from the statement

```
SELECT   *
FROM     Student
ORDER BY Address DESC, Name ASC;
```

Due to their different semantics (try it out and see!), both commands yield different table results.

Further, you will need information about the *like* command for string pattern search. Please look it up online in Oracle manuals or SQL tutorials when the command is needed.

A few queries will require that you have an online look into SQL manuals to obtain a missing component. But in the end, these queries are not difficult.

The basis of this question is a data set about a company's processed orders and sales of toy cars (and not real cars) from the following Kaggle web page. You do not have to (but can) download the dataset since everything you need will be prepared for you.

(a) [5 points] Perform the following tasks and follow the instructions in every detail.

Task 1 Create a table schema for this dataset. To achieve this, do the following:

(1) The Kaggle web page provides a table with a description of all attributes of the dataset as well as their meaning. Study this information.
(2) We will *not* consider the information about status, MSRP, product code, and phone from the dataset since we assume it is not needed for our (hypothetical) applications.
(3) Create the table schema *ToyCarOrders* by taking the remaining attributes and selecting meaningful data types for them.
(4) Nouns and adjectives in attribute names should begin with an uppercase letter and have lowercase letters otherwise (example: *ORDERDATE* $\rightarrow$ *OrderDate*). Underscores should be removed (example: *DAYS_SINCE_LASTORDER* $\rightarrow$ *DaysSinceLastOrder*).
(5) Think about which attributes should get the `NOT NULL` constraint, that is, should be disallowed to have the `NULL` value.
(6) Data type names have to be completely written in uppercase letters (e.g., `INT`).
(7) The first seven attributes *OrderNumber*, *QuantityOrdered*, *PriceEach*, *OrderLineNumber*, *Sales*, *OrderDate*, and *DaysSinceLastOrder* should have a non-string (that is, not a `VARCHAR`) data type. The remaining nine attributes *ProductLine*, *CustomerName*, *AddressLine1*, *City*, *PostalCode*, *Country*, *ContactLastName*, *ContactFirstName*, and *DealSize* should have a string data type. Note that Oracle prefers `VARCHAR2` instead of `VARCHAR` as string data type.
(8) Select an appropriate primary key for the schema and give reasons for your choice.
(9) Show the execution of the creation command in Oracle SQL Developer.

Task 2 Show that the table created is empty at this point. Note that all SQL commands have to be completely written in uppercase letters to increase readability of queries.

Task 3 Copy, paste, and execute all `insert` commands from the provided ASCII file *ToyCarOrders Insert Commands.sql* in the upper window of Oracle SQL Developer.

Task 4 Show that and how the table is filled with these tuples.

For each task, provide an appropriate screenshot that shows the SQL command or SQL query as well as its execution result (see example above). Show all `insert` commands with a few "`1 row inserted`" lines. The right sides of commands and results can be incomplete in the screenshots.

In the following, design and execute SQL queries for some colloquial queries and tasks.

(b) [3 points] Provide an organized view of the table to obtain the best overview of the orders. Give a brief reason for your query.

(c) [3 points] Find the quantities ordered, the prices for each unit, and the sales values for order 10109 with orderline number 4, order 10100 with orderline number 2, and order 10101 with orderline number 1. Output all five attributes mentioned.

(d) [3 points] Determine which product lines can be found in the table.

(e) [3 points] Determine how many different product lines can be found in the table and keep the result under the new attribute "Total Number of Product Lines".

(f) [3 points] Determine the dates of order lines with a unit price between $77.77 and $111.11. Arrange the dates in descending order.

(g) [3 points] The sales values in the table are the result of the product of the price of each unit and the quantity ordered. Check for each tuple that the stored sales values are correct. For this, show the stored sales value, the computed sales value, and their difference (that should be 0 obviously) for each tuple.

(h) [3 points] Find the numbers of orders where the first name of the contact person begins with "Ro". Output the first name of the contact person too.

(i) [3 points] Compute the average sales value, the minimum sales value, the maximum sales value, and the sum of all sales of those order lines that do not have customers from Germany. Provide meaningful names for the computed values. Round each value to two decimal places.

(j) [3 points] Determine how many days have passed between the oldest order and the most recent order.