Directions:

* You may use drawings, tables, and figures as a part of your answers
* You may use bulleted phrases or complete paragraphs in your answers
* Make sure you answer the question
* Your work should be done using MySQL Workbench, MS Word, and a drawing tool such as Visio
  + Nothing that is handwritten or drawn by hand will be graded.
  + No pictures will be graded.
  + Screenshots of MySQL Workbench are needed for some problems
    - For those problems, only take a screenshot of the relevant portions of the screen, not a screenshot of the entire screen
* You will need to use the my\_guitar\_shop ERD and database for many of the problems on this exam
  + Be sure to rerun the SQL script that builds the my\_guitar\_shop database before starting this exam. The exam is written assuming you have a clean copy of the database.
* You will need to submit this document and the SQL answer sheet to Canvas when you are finished
* For relational algebra problems:
  + You must follow all guidance regarding relational algebra that has been given throughout the quarter
* For SQL problems:
  + You must follow all guidance regarding SQL that has been given throughout the quarter
  + For all questions requiring a SQL statement, you must submit the statement twice, once as a screentshot in this document and once in the .sql answer sheet.
  + You must also follow all guidance regarding formatting that has been given throughout the quarter

5-point questions

# What is(are) the name(s) of the foreign key constraint(s) in the my\_guitar\_shop.order\_lines table?

order\_id: is part of the Primary Key and Foreign Key, and is Not Null, thus this FK will have the Not Null constraint, Key constraint, and the Referential Integrity because of those attributes

product\_id: Non-prime attribute, part of the Foreign Key, and is Not Null, thus this FK will have the Not Null constraint and the Referential Integrity because of those attributes

# Write the relational notation for the instance of the my\_guitar\_shop.product\_locations table.

Instance

product\_locations(*prod\_id*, *wh\_id*, qty\_on\_hand, eoq)

# Normalize the 1NF relation 3NF showing the resulting 2NF and 3NF relations. Show your work in relational notation.

| **1NF Relation** | **2NF relation** | **3NF relation** |
| --- | --- | --- |
| R(A,B,C,D,E,F,G,H,I,J)  FD1 G -> D,E  FD2 I -> J  FD3 (A,B,C) -> G  FD4 (A,B) -> H,I  FD5 E -> F | R1(A,B,C,D,E,F,G)  R2(A,B,H,I,J) | R1a(A,B,C,G)  R1b(D,E,G)  R1c(E,F)  R2a(A,B,H,I)  R2b(I,J) |

# Consider the following relations for a database that keeps track of orders:

vendor(vendorID, vendorStAddr,vendorCity,vendorState,vendorZip,vendorPhone)

invoice(invoiceID,date,*clientID*,terms,amount,*vendorID*)

client(clientID,clientStAddr,clientCity,clientState,clientZip,clientPhone)

List the foreign keys that are in these relations (*The FKs are not currently identified*). Do not list any attributes that you believe are foreign keys to relations that are not listed above.

There are two relationships in the above relations. One of the relationships is between vendor and invoice, and the second is between client and invoice.

Specify the attributes that would be foreign keys in these relations.

* Do not assume any attributes are foreign keys to relations that are not listed above.
* Show your answer as relation(*attributeName*) for as many foreign keys as are necessary.
* Do not list any attributes besides the foreign keys

invoice(*venderID, clientID)*

# A diagram of a restaurant Description automatically generated with medium confidenceIdentify at least four business rules for the following ER diagram.

**restaurant and menuItems**

**A restaurant can have one to many menu items.**

**A menu item has one and only one restaurant.**

**menu item and options**

**A menu item can have zero to many options.**

**A option has one and only one menu item.**

# Using the figure in previous problem, list each entity and state what foreign key, if any, would be within the entity to form the relationships shown.

menuItems(*restaurantID*)

options(*itemID*)

10-Point Questions

# Write the relational algebra expressions that will list each warehouse name and the names of the products that are in the warehouse’s inventory. Insert rows into the table as necessary.

|  |  |
| --- | --- |
| PRODUCT\_LOCATION(\*) | RESULT1⟵ σ(qty\_on\_hand > 0)(PRODUCT\_LOCATION) |
| PRODUCTS(\*)  PRODUCTS\_LOCATIONS(\*) | RESULT2⟵ RESULT1⨝product\_id = prod\_id PRODUCTS |
| PRODUCTS(\*)  PRODUCTS\_LOCATIONS(\*)  WAREHOUSES(\*) | RESULT3⟵ RESULT2⨝wh\_id = warehouse\_idWAREHOUSES |
| PRODUCTS(product\_name)  WAREHOUSES(wh\_name) | RESULT⟵ π(wh\_name , product\_name)RESULT3 |

# Write the SQL statement that will implement the problem described in the previous question.

A screenshot of a computer

Description automatically generated with medium confidence

1. Write the plain English for what the following relational algebra expressions will do.

|  |
| --- |
| RESULT1 ⟵σ(product\_name = ‘Gibson SG’)(PRODUCTS) |
| RESULT2 ⟵ RESULT1⨝product\_id=product\_idORDER\_LINES |
| RESULT3 ⟵ π (order\_id)(RESULT2) |
|  |
| RESULT4 ⟵σ(product\_name = ‘Fendor Stratocaster’)(PRODUCTS) |
| RESULT5 ⟵ RESULT4⨝product\_id=product\_idORDER\_LINES |
| RESULT6 ⟵ π (order\_id)(RESULT5) |
|  |
| RESULT7 ⟵ RESULT3 ⋂ RESULT6 |
| RESULT8 ⟵ RESULT7⨝order\_id=order\_idORDERS |
| RESULT9 ⟵ RESULT8⨝customer\_id=customer\_idCUSTOMERS |
| RESULT ⟵ π (first\_name,last\_name)(RESULT9) |

List the first and last name of any customer that ordered a Gibson SG and a Fendor Stratocaster

# Write the SQL statement that will implement the relational algebra expressions in the previous problem.

A screenshot of a computer

Description automatically generated

# Write the plain English for the following SQL statement

SELECT product\_name

FROM products

WHERE product\_id NOT IN (

SELECT product\_id

FROM order\_lines

);

List the name of any product that has not been orderd before.

# Using a minus, write the relational algebra expressions for the SQL statement in the previous problem. Add rows to the table as necessary.

|  |  |
| --- | --- |
| PRODUCTS(product\_id) | RESULT1⟵ π(product\_id)( PRODUCTS) |
| ORDER\_LINES(product\_id) | RESULT2⟵ π(product\_id)( ORDER\_LINES) |
| PRODUCTS(product\_id) | RESULT3⟵ RESULT1- RESULT2 |
| PRODUCTS(\*) | RESULT4⟵RESULT3⨝product\_id=product\_id PRODUCTS |
| PRODUCTS(product\_name) | RESULT⟵ π(product\_name)(RESULT4) |

# Write a SQL statement that will list the first and last names of the two customers that have placed the most orders. An order is counted as one order, regardless of how many products are on the order.

A screenshot of a computer

Description automatically generated with medium confidence