VHT2 — VHT2 TASK 1: NORMALIZATION AND DATABASE DESIGN

DATA MANAGEMENT - APPLICATIONS — C170
PRFA — VHT2

TASK OVERVIEW SUBMISSIONS EVALUATION REPORT

COMPETENCIES

4018.1.1 : Conceptual Models to Physical Schemas
The graduate creates conceptual data models and translates them into physical schemas.

4018.1.2 : Create Databases
The graduate creates databases utilizing SQL Data Definition Language (DDL) in MySQL environment.

4018.1.3 : Create/Modify Tables and Views
The graduate creates and modifies tables and views employing SQL Data Definition Language (DDL) in MySQL environment.

4018.1.4 : Create Primary Keys/Foreign Keys and Indexes
The graduate creates and modifies primary keys (PKs) and foreign keys (FKs) and indexes with SQL Data Definition Language (DDL) in MySQL environment.

4018.1.5 : Populate Tables
The graduate populates tables with insert, update, and delete using DML in the MySQL environment.

4018.1.6 : Create Simple and Complex Queries
The graduate creates simple Select-From-Where (SFW) and complex 3+ table join queries with Data Manipulation Language (DML) in MySQL environment.

INTRODUCTION

For this assessment, you will be creating an entity-relationship (ER) model, databases, tables, and queries for two fictional small businesses. To complete this assessment, you will use SQL to test and run a database application that you will develop. After running the code, you will take a screenshot of your results and paste the screenshot into a document that you will submit.

The work you complete for each part of the assessment (i.e., the design models and diagrams, tables, written explanations, SQL script code, and screenshot results from running your SQL scripts in a SQL tool) should be saved as a single PDF file that you will submit.

Note: If you do not have access to a database tool, you may use SQL Fiddle (an online SQL tool) to complete this assessment. The tool can be accessed using the "SQL Fiddle" link in the Web Links section of this task. Instructions for how to use SQL Fiddle for each part of the assessment are included in the attached document "SQL Fiddle Instructions." Please note that for each part of the assessment, there are explicit instructions on what SQL code you will need to copy and paste into the SQL Fiddle panels to run your test.

SCENARIO

You are a database designer and developer who has been hired by two local businesses, Nora's Bagel Bin and Jaunty Coffee Co., to build databases to help them manage their businesses. First, you will design a normalized physical database model to store data for Nora's Bagel Bin's ordering system. Then, you will use an existing database design document for Jaunty Coffee Co. to create its database. Once the tables have been built, you will load them with sample data and create a view and an index to protect and improve query performance. Finally, you will create both a simple query and a more complex table joins query to produce meaningful reports from the newly created database.

REQUIREMENTS

Your submission must be your original work. No more than a combined total of 30% of a submission and no more than a 10% match to any one individual source can be directly quoted or closely paraphrased from sources, even if cited correctly. An originality report is provided when you submit your task that can be used as a guide.

You must use the rubric to direct the creation of your submission because it provides detailed criteria that will be used to evaluate your work. Each requirement below may be evaluated by more than one rubric aspect. The rubric aspect titles may contain hyperlinks to relevant portions of the course.

*Tasks may **not** be submitted as cloud links, such as links to Google Docs, Google Slides, OneDrive, etc., unless specified in the task requirements. All other submissions must be file types that are uploaded and submitted as attachments (e.g., .docx, .pdf, .ppt).*

A. Construct a normalized physical database model to represent the ordering process for Nora's Bagel Bin by doing the following:

Note: Before proceeding, familiarize yourself with the ordering process for Nora's Bagel Bin by reviewing the following documents in the Supporting Documents section of this task: the shop's unnormalized sales order form ("Bagel Order Form") and the first normal form (1NF) provided in the "Nora's Bagel Bin Database Blueprints."

1. Complete the second normal form (2NF) section of the attached "Nora's Bagel Bin Database Blueprints" document by doing the following:

a. Assign *each* attribute from the 1NF table into the correct 2NF table.
b. Describe the relationship between the **two** pairs of 2NF tables by indicating their cardinality in *each* of the dotted cells: one-to-one (1:1), one-to-many (1:M), many-to-one (M:1), or many-to-many (M:M).
c. Explain how you assigned attributes to the 2NF tables and determined the cardinality of the relationships between your 2NF tables.

2. Complete the third normal form (3NF) section of the attached "Nora's Bagel Bin Database Blueprints" document by doing the following:
a. Assign *each* attribute from your 2NF "Bagel Order" table into one of the new 3NF tables. Copy *all* other information from your 2NF diagram into the 3NF diagram.
b. Provide *each* 3NF table with a name that reflects its contents.
c. Create a new field that will be used as a key linking the **two** 3NF tables you named in part A2b. Ensure that your primary key (PK) and foreign key (FK) fields are in the correct locations in the 3NF diagram.
d. Describe the relationships between the 3NF tables by indicating their cardinality in *each* of the dotted cells: one-to-one (1:1), one-to-many (1:M), many-to-one (M:1), or many-to-many (M:M).
e. Explain how you assigned attributes to the 3NF tables and determined the cardinality of the relationships between your 3NF tables.
3. Complete the "Final Physical Database Model" section of the attached "Nora's Bagel Bin Database Blueprints" document by doing the following:
a. Copy the table names and cardinality information from your 3NF diagram into the "Final Physical Database Model" and rename the attributes.
b. Assign **one** of the following **five** data types to *each* attribute in your 3NF tables: CHAR(), VARCHAR(), TIMESTAMP, INTEGER, or NUMERIC(). *Each* data type must be used *at least* once.

B. Create a database using the attached "Jaunty Coffee Co. ERD" by doing the following:
1. Develop SQL code to create *each* table as specified in the attached "Jaunty Coffee Co. ERD" by doing the following:
a. Provide the SQL code you wrote to create *all* the tables.
b. Demonstrate that you tested your code by providing a screenshot showing your SQL commands and the database server's response.
2. Develop SQL code to populate *each* table in the database design document by doing the following:

Note: This data is not provided. You will be fabricating the data for this step.

a. Provide the SQL code you wrote to populate the tables with *at least* **three** rows of data in *each* table.
b. Demonstrate that you tested your code by providing a screenshot showing your SQL commands and the database server's response.
3. Develop SQL code to create a view by doing the following:
a. Provide the SQL code you wrote to create your view. The view should show *all* of the information from the "Employee" table but concatenate *each* employee's first and last name, formatted with a space between the first and last name, into a new attribute called employee_full_name.
b. Demonstrate that you tested your code by providing a screenshot showing your SQL commands and the database server's response.

4. Develop SQL code to create an index on the coffee_name field by doing the following:
   a. Provide the SQL code you wrote to create your index on the coffee_name field from the "Coffee" table.
   b. Demonstrate that you tested your code by providing a screenshot showing your SQL commands and the database server's response.
5. Develop SQL code to create an SFW (SELECT–FROM–WHERE) query for *any* of your tables or views by doing the following:
   a. Provide the SQL code you wrote to create your SFW query.
   b. Demonstrate that you tested your code by providing a screenshot showing your SQL commands and the database server's response.
6. Develop SQL code to create a query by doing the following:
   a. Provide the SQL code you wrote to create your table joins query. The query should join together **three** different tables and include attributes from *all* three tables in its output.
   b. Demonstrate that you tested your code by providing a screenshot showing your SQL commands and the database server's response.

C. Submit parts A and B as a PDF, with *each* part clearly labeled.

D. Demonstrate professional communication in the content and presentation of your submission.

File Restrictions

File name may contain only letters, numbers, spaces, and these symbols: ! - _ . * ' ( )
File size limit: 200 MB
File types allowed: doc, docx, rtf, xls, xlsx, ppt, pptx, odt, pdf, txt, qt, mov, mpg, avi, mp3, wav, mp4, wma, flv, asf, mpeg, wmv, m4v, svg, tif, tiff, jpeg, jpg, gif, png, zip, rar, tar, 7z

RUBRIC

A1A:1NF TABLE ATTRIBUTES TO 2NF TABLE

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not assign *any* attributes from the 1NF table into the correct 2NF table.	The submission assigns *at least* 1 but not *all* attributes from the 1NF table into the correct 2NF table.	The submission assigns *all* attributes from the 1NF table into the correct 2NF table.

A1B:RELATIONSHIPS BETWEEN 2NF TABLES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT

The submission does not describe the relationship between the 2 pairs of 2NF tables by indicating their cardinality in *each* of the dotted cells.

The submission does not correctly describe the relationship between the 2 pairs of 2NF tables by indicating their cardinality in *each* of the dotted cells.

The submission correctly describes the relationship between the 2 pairs of 2NF tables by indicating their cardinality in *each* of the dotted cells.

A1C:EXPLANATION OF ATTRIBUTES AND CARDINALITY OF 2NF TABLES

NOT EVIDENT

The submission does not explain how the attributes were assigned to the 2NF tables and how the cardinality of the relationships between the 2NF tables was determined.

APPROACHING COMPETENCE

The submission illogically explains how the attributes were assigned to the 2NF tables or how the cardinality of the relationships was determined between the 2NF tables.

COMPETENT

The submission logically explains how the attributes were assigned to the 2NF tables and how the cardinality of the relationships was determined between the 2NF tables.

A2A:2NF TABLE ATTRIBUTES TO 3NF TABLE

NOT EVIDENT

The submission does not assign *any* attributes from the 2NF "Bagel Order" table into one of the new 3NF tables.

APPROACHING COMPETENCE

The submission does not assign 1 or more attributes from the 2NF "Bagel Order" table into one of the new 3NF tables or copies only *some* of the other information from the 2NF diagram into the 3NF diagram.

COMPETENT

The submission assigns *all* attributes from the 2NF "Bagel Order" table into one of the new 3NF tables and copies *all* other information from the 2NF diagram into the 3NF diagram.

A2B:NAMING OF TABLES

NOT EVIDENT

The submission does not provide a name for *any* 3NF table.

APPROACHING COMPETENCE

The submission provides a name for 1 or more 3NF tables, but the name for 1 or more tables does not reflect their contents.

COMPETENT

The submission provides a name for *each* 3NF table that reflects its contents.

A2C:3NF KEYS BETWEEN TABLES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not provide a new field that will be used as a key linking the 2 3NF tables.	The submission creates a new field that will be used as a key linking the 2 3NF tables, but the PK and FK fields are not in the correct locations in the 3NF diagram.	The submission creates a new field that will be used as a key linking the 2 3NF tables. The PK and FK fields are in the correct locations in the 3NF diagram.

A2D:RELATIONSHIPS BETWEEN 3NF TABLES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not describe the relationships between the 3NF tables by indicating their cardinality.	The submission does not correctly describe the relationships between the 3NF tables by indicating their cardinality in *each* of the dotted cells.	The submission correctly describes the relationships between the 3NF tables by indicating their cardinality in *each* of the dotted cells.

A2E:EXPLANATION OF ATTRIBUTES AND CARDINALITY OF 3NF TABLES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not explain how the attributes were assigned to the 3NF tables and how the cardinality of the relationships between the 3NF tables was determined.	The submission illogically explains how the attributes were assigned to the 3NF tables or how the cardinality of the relationships was determined between the 3NF tables.	The submission logically explains how the attributes were assigned to the 3NF tables and how the cardinality of the relationships was determined between the 3NF tables.

A3A:COPYING INFORMATION AND RENAMING ATTRIBUTES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not copy *any* table names and cardinality information from the 3NF diagram into the "Final Physical	The submission incorrectly copies 1 or more table names or cardinality information from	The submission correctly copies the table names and cardinality information from the 3NF diagram into the "Final Physical

Database Model" or does not rename the attributes.

the 3NF diagram into the "Final Physical Database Model" or incorrectly renames 1 or more attributes.

Database Model" and correctly renames the attributes.

A3B:FIVE DATA TYPES

NOT EVIDENT

The submission does not assign *any* of the 5 data types to *any* attribute in the 3NF tables.

APPROACHING COMPETENCE

The submission assigns 1 of the 5 data types to *at least* 1 of the attributes in the 3NF tables, but a data type is not assigned to *each* of the attributes. Or *each* data type is not used *at least* once.

COMPETENT

The submission assigns 1 of the 5 data types to *each* attribute in the 3NF tables, and *each* data type is used *at least* once.

B1A:SQL CODE: TABLES

NOT EVIDENT

The submission does not provide the SQL code written for *any* table.

APPROACHING COMPETENCE

The submission provides inaccurate or illogical SQL code written for 1 or more tables.

COMPETENT

The submission provides accurate and logical SQL code written for *all* the tables.

B1B:TABLES: SCREENSHOT OF SQL CODE RESULTS

NOT EVIDENT

The submission does not demonstrate that the code was tested by providing a screenshot showing the SQL commands and the database server's response.

APPROACHING COMPETENCE

Not applicable.

COMPETENT

The submission demonstrates that the code was tested by providing a screenshot showing the SQL commands and the database server's response.

B2A:SQL CODE: DATA POPULATION

NOT EVIDENT

APPROACHING

COMPETENT

The submission does not provide the SQL code written to populate the tables.

COMPETENCE

The submission provides inaccurate or illogical SQL code written to populate the tables, or 1 or more tables does not include *at least* 3 rows of data.

The submission provides accurate and logical SQL code written to populate the tables with *at least* 3 rows of data in *each* table.

B2B:DATA: SCREENSHOT OF SQL CODE RESULTS

NOT EVIDENT

The submission does not demonstrate that the code was tested by providing a screenshot showing the SQL commands and the database server's response.

APPROACHING COMPETENCE

Not applicable.

COMPETENT

The submission demonstrates that the code was tested by providing a screenshot showing the SQL commands and the database server's response.

B3A:SQL CODE: VIEW FOR EMPLOYEE INFORMATION

NOT EVIDENT

The submission does not provide the SQL code written to create the view.

APPROACHING COMPETENCE

The submission provides inaccurate or illogical SQL code written to create the view. Or the view does not show *all* of the information from the "Employee" table. Or the view does not concatenate *each* employee's first and last name, with a space between the first and last name, into an attribute called employee_full_name.

COMPETENT

The submission provides accurate and logical SQL code written to create the view. The view shows *all* of the information from the "Employee" table and concatenates *each* employee's first and last name, with a space between the first and last name, into an attribute called employee_full_name.

B3B:VIEW: SCREENSHOT OF SQL CODE RESULTS

NOT EVIDENT

The submission does not demonstrate that the code was tested by providing a screen-

APPROACHING COMPETENCE

Not applicable.

COMPETENT

The submission demonstrates that the code was tested by providing a screenshot showing the

shot showing the SQL commands and the database server's response.		SQL commands and the database server's response.

B4A:SQL CODE: INDEX OF COFFEE INFORMATION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not provide the SQL code written to create an index.	The submission provides inaccurate or illogical SQL code written to create an index on the coffee_name field from the "Coffee" table.	The submission provides accurate and logical SQL code written to create the index for the coffee_name field of the "Coffee" table.

B4B:INDEX: SCREENSHOT OF SQL CODE RESULTS

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not demonstrate that the code was tested by providing a screenshot showing the SQL commands and the database server's response.	Not applicable.	The submission demonstrates that the code was tested by providing a screenshot showing the SQL commands and the database server's response.

B5A:SQL CODE: SFW QUERIES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not provide SQL code written to create the SFW query.	The submission provides inaccurate or illogical SQL code written to create the SFW query on *any* of the tables or views.	The submission provides accurate and logical SQL code written to create the SFW query on *any* of the tables or views.

B5B:SFW QUERIES: SCREENSHOT OF SQL CODE RESULTS

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not demonstrate that the code was tested by providing a screen-	Not applicable.	The submission demonstrates that the code was tested by providing a screenshot showing the

shot showing the SQL commands and the database server's response.		SQL commands and the database server's response.

B6A:SQL CODE: JOIN QUERY

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not provide the SQL code written to create the table joins query.	The submission provides inaccurate or illogical SQL code written to create the table joins query. Or the table joins query does not join together 3 different tables and include attributes from *all* 3 tables in its output.	The submission provides accurate and logical SQL code written to create the table joins query. The query joins together 3 different tables and includes attributes from *all* 3 tables in its output.

B6B:JOIN QUERY: SCREENSHOT OF SQL CODE RESULTS

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not demonstrate that the code was tested by providing a screenshot showing the SQL commands and the database server's response.	Not applicable.	The submission demonstrates that the code was tested by providing a screenshot showing the SQL commands and the database server's response.

C:PDF SUBMISSION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The submission does not provide a PDF of parts A and B.	The submission provides a PDF of parts A and B, but *each* part is not clearly labeled.	The submission provides a PDF of parts A and B with *each* part clearly labeled.

D PROFESSIONAL COMMUNICATION

NOT EVIDENT	APPROACHING	COMPETENT

Content is unstructured, is disjointed, or contains pervasive errors in mechanics, usage, or grammar. Vocabulary or tone is unprofessional or distracts from the topic.

COMPETENCE

Content is poorly organized, is difficult to follow, or contains errors in mechanics, usage, or grammar that cause confusion. Terminology is misused or ineffective.

Content reflects attention to detail, is organized, and focuses on the main ideas as prescribed in the task or chosen by the submission. Terminology is pertinent, is used correctly, and effectively conveys the intended meaning. Mechanics, usage, and grammar promote accurate interpretation and understanding.

WEB LINKS

SQL Fiddle

SUPPORTING DOCUMENTS

Bagel Order Form.docx

Jaunty Coffee Co. ERD.xlsx

Nora's Bagel Bin Database Blueprints.docx

SQL_Fiddle_Instructions.docx