**BUS5DWR – Data Wrangling**

**Student Name: Sathavara Suchi**

**Student ID: 22237665**

**Assignment 01: Database design and SQL**

**Marks: 60 marks (30% of the final grade)**

**Assignment Type: Individual**

|  |  |
| --- | --- |
| **SILOS** | **SILO 1:** Design, formulate and compose database schema and query structures for analytics initiatives |
|  | **SILO 3:** Construct SQL and computer programs to wrangle data. |

# Overview

Over the past few weeks, you have gained a general understanding of designing databases and how to use SQL to retrieve information from business data. This assignment allows you to demonstrate what you have learned.

Academic Integrity

**Plagiarism** occurs when you use words, ideas, or work products attributable to another identifiable person or source:

* without attributing the work to the source from which it was obtained
* in a situation in which there is a legitimate expectation of original authorship
* in order to obtain some benefit, credit, or gain which need not be monetary

**Collusion** is a form of cheating which occurs when people work together in a deceitful way to develop a submission for an assessment which has been restricted to individual effort.

**By submitting\* this piece of work and signing this document, I declare that:**

* 1. The work is my own individual work.
  2. I have not previously submitted all or part of this work for assessment in any subject, unless the subject coordinator for the current subject (or my research supervisor, if applicable) has given me written permission to reuse specific material and I have correctly referenced the material taken from my own earlier work.
  3. I have read and agree to be bound by the Statutes, Regulations and Policies of the University relating to Academic Integrity available at [http://www.latrobe.edu.au/students/academic-integrity;](http://www.latrobe.edu.au/students/academic-integrity) and
  4. I may be subject to student discipline processes in the event of an act of academic misconduct by me including an act of plagiarism or cheating.

I further grant to the University or any third party authorised by the University (www.latrobe.edu.au/text-match) the right to reproduce and/or communicate (make available online or electronically transmit) the work I have submitted for the purpose of detecting plagiarism.

# Assignment Description

**Part 1: Database design (27 marks)**

**Question 1**

**Urban Pedals: Revamping the Bike Rental Database for Operational Efficiency**

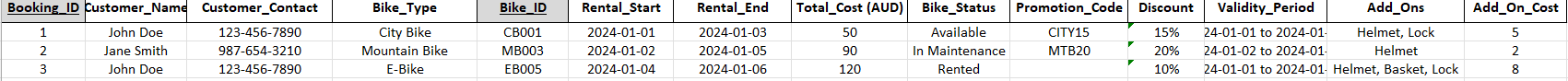
The Johnson family runs Urban Pedals, a bike rental service offering city bikes, mountain bikes, and e-bikes for durations ranging from a few hours to several days. To enhance customer experience, they provide optional add-ons like helmets, locks, and baskets at fixed costs.

To attract more customers, Urban Pedals introduced bike-specific discount code promotions, such as **CITY15** for 15% off city bikes or **MTB20** for 20% off mountain bikes during specific periods. Customers must enter the promo code at booking to receive the discount, which applies only to bike rentals, not add-ons. Add-ons are charged separately, and customers can select multiple add-ons per booking.

The current design also makes it difficult to analyse the effectiveness of promotions or track revenue generated from specific bike types and add-ons. As Urban Pedals continues to grow, managing data efficiently has become essential for streamlining operations and supporting business decisions.

Currently, the database consists of a single table where all the information stored in a flat structure. Here’s a sample of how the current table looks (Primary key is consist of all the underlined attributes):

**Question 1.1 (7.5 marks)**



1. Is this table satisfy 1NF, 2NF, and 3NF? Explain why. **(4.5 marks)**

**Answer:**

|  |  |  |
| --- | --- | --- |
| **Normal Form** | **Satisfy**  **(Yes/No)** | **Explanation** |
| **1 NF** | **No** | **Column Add-on doesn’t contain atomic value. Column Validity\_Period contains multiple values.** |
| **2 NF** | **No** | **It is not 1NF.** |
| **3 NF** | **No** | **It is not 2NF.** |

1. Give one example for each insertion anomaly, deletion anomaly and update anomaly that may happen when using this table to record data. **(3 marks)**

**Answer:**  Example of insertion, deletion and update anomaly

* Insertion Anomaly: If there is new bike in the market but hasn’t been rented than we can’t add information about that. we can’t insert details about new bike without putting booking record for it.
* Deletion Anomaly: If we delete booking of specific bike type in one row we will lose information about that bike type entirely.
* Update Anomaly: If we want to update discount which is applied to one specific type of bike then we have to update every row which contains that promotion code.

**Question 1.2 (19.5 marks)**

The family has asked for your help to redesign the database to improve its efficiency. Your task is to normalize the database to at least **3NF** and create an **Entity-Relationship Diagram (ERD)** that better supports the business operations.

**NOTE**:

* *The given table doesn’t capture all the entities and attributes required in the final design. Therefore, you will need to identify the relevant entities and their attributes to fully normalize the database.*
* *You can assume that maximum of one promotion code can be applied to a booking*

Using [https://erdplus.com](https://erdplus.comL) **(mandatory)**, design a database in the normalised form. You are asked to:

1. Present your entity-relationship diagram (ERD). Any assumption should be stated clearly **(12 Marks)**.
2. Present the generated relational schema. The primary key and possible foreign keys of each table should be shown **(4 marks)**.
3. Present the generated SQL statements to compose the tables with the appropriate data type set for each attribute **(3.5 marks).**

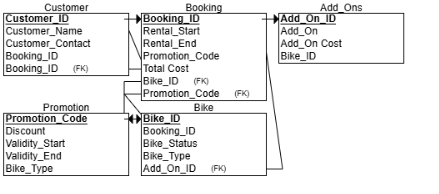
**Answer:**

1. E-R Diagram:

A diagram of a company

Description automatically generated

2.) Schema:



3) SQL STATEMENTS:

CREATE TABLE Add\_Ons

(

Add\_On\_ID INT NOT NULL,

Add\_On VARCHAR(50) NOT NULL,

Add\_On\_Cost INT NOT NULL,

Bike\_ID VARCHAR(50) NOT NULL,

PRIMARY KEY (Add\_On\_ID)

);

CREATE TABLE Promotion

(

Discount FLOAT NOT NULL,

Validity\_Start DATE NOT NULL,

Validity\_End DATE NOT NULL,

Promotion\_Code VARCHAR(50) NOT NULL,

Bike\_Type VARCHAR(50) NOT NULL,

PRIMARY KEY (Promotion\_Code)

);

CREATE TABLE Bike

(

Booking\_ID INT NOT NULL,

Bike\_ID VARCHAR(50) NOT NULL,

Bike\_Status VARCHAR(50) NOT NULL,

Bike\_Type VARCHAR(50) NOT NULL,

Add\_On\_ID INT NOT NULL,

PRIMARY KEY (Bike\_ID),

FOREIGN KEY (Add\_On\_ID) REFERENCES Add\_Ons(Add\_On\_ID)

);

CREATE TABLE Booking

(

Booking\_ID INT NOT NULL,

Rental\_Start DATE NOT NULL,

Rental\_End DATE NOT NULL,

Promotion\_Code FLOAT NOT NULL,

Total\_Cost INT NOT NULL,

Bike\_ID VARCHAR(50) NOT NULL,

Promotion\_Code VARCHAR(50) NOT NULL,

PRIMARY KEY (Booking\_ID),

FOREIGN KEY (Bike\_ID) REFERENCES Bike(Bike\_ID),

FOREIGN KEY (Promotion\_Code) REFERENCES Promotion(Promotion\_Code)

);

CREATE TABLE Customer

(

Customer\_ID INT NOT NULL,

Customer\_Name VARCHAR(50) NOT NULL,

Customer\_Contact VARCHAR(30) NOT NULL,

Booking\_ID INT NOT NULL,

Booking\_ID INT NOT NULL,

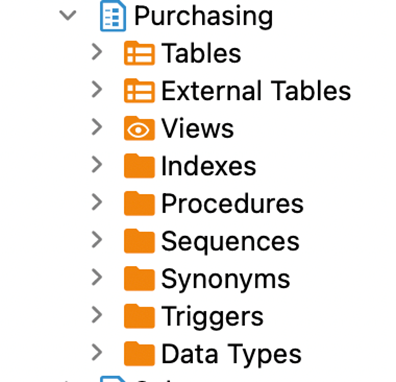
PRIMARY KEY (Customer\_ID),

FOREIGN KEY (Booking\_ID) REFERENCES Booking(Booking\_ID)

);

**Part 2: Database retrieval using SQL (33 Marks)**

We will make use of the data contained in bus5dwr.Purchasing schema within the lmban-teaching.database.windows.net. Please refer to Workshop 1 for instructions on how to access the server via DBeaver if you have not accessed it already.



Start by exploring the contents of the tables to understand the meaning of the attributes of each table.

**Question 2.1** The manager wants a better insight into the company business to support his future decisions. He requests the IT department to provide answers to the following questions. You are asked to construct SQL code to find the answer for each question.

Fill in below the code ***in text*** and screenshot the result table highlighting the columns (adjust your results section to show first 8 rows and **must show the number of fetched rows and time**). Result of one question must be shown in only one return table. In the last three questions, you may support your criteria with one or more SQL statements with screenshots.

***Example (No marks)***

|  |  |
| --- | --- |
| ***Question*** | Display the orders that have an order quantity greater than 3. |
| ***Code*** | **SELECT** pod.PurchaseOrderID, pod.ProductID , pod.OrderQty  **FROM** Purchasing.PurchaseOrderDetail pod  **WHERE** pod.OrderQty > 3 |
| ***Returned table*** |  |

**2.1  *(2 Marks)***

|  |  |
| --- | --- |
| ***Question*** | List all vendors who are actively supplying products to the company. Include their name, account number, and credit rating. Ensure the results are organized alphabetically by vendor name. *Hint: Use active flag value to check whether vendor is active. Likewise, you will need to figure out appropriate columns to answer the following questions.* |
| ***Code*** | **SELECT** *v*.Name, *v*.AccountNumber, *v*.CreditRating  **FROM** Purchasing.Vendor *v*  **WHERE**  *v*.ActiveFlag = 1  **ORDER** **BY**  *v*.Name **ASC** |
| ***Returned table*** |  |

**2.2  *(2 Marks)***

|  |  |
| --- | --- |
| ***Question*** | Find all vendors whose names start with "sport" and who are currently active suppliers. Arrange the results by their name in alphabetical order. |
| ***Code*** | **SELECT** *v*.Name **FROM** Purchasing.Vendor *v*  **WHERE**  *v*.Name **LIKE** 'sport%' **AND** *v*.ActiveFlag = 1 **ORDER** **BY** *v*.Name **ASC** |
| ***Returned table*** |  |

**2.3 *(2 Marks)***

|  |  |
| --- | --- |
| ***Question*** | Identify the affordable products supplied by preferred vendors and where the last receipt cost was less than $40. Display the product id, vendor name, and last receipt cost details, sorted by the cost in descending order. |
| ***Code*** | **SELECT**  pv.ProductID,  *v*.Name **AS** *VendorName*,  *pv*.LastReceiptCost  **FROM**  Purchasing.ProductVendor *pv*  **JOIN**  Purchasing.Vendor *v* **ON** *pv*.BusinessEntityID = *v*.BusinessEntityID  **WHERE**  *v*.PreferredVendorStatus = 1  **AND** *pv*.LastReceiptCost < 40 -- less than 40  **ORDER** **BY**  *pv*.LastReceiptCost **DESC**; |
| ***Returned table*** |  |

**2.4 *(3 Marks)***

|  |  |
| --- | --- |
| ***Question*** | How many distinct orders were made in 2014 that were shipped using either “CARGO” or “TRUCK”? Display the ship method name, total order quantity and the total number of distinct orders, sorted by the order quantity in descending order. |
| ***Code*** | **SELECT**  *sm*.Name **AS** *ShipMethodName*,  **SUM**(*pod*.OrderQty) **AS** *TotalOrderQuantity*,  **COUNT**(**DISTINCT** *poh*.PurchaseOrderID) **AS** *DistinctOrders*  **FROM**  Purchasing.PurchaseOrderHeader *poh*  **JOIN**  Purchasing.ShipMethod *sm* **ON** *poh*.ShipMethodID = *sm*.ShipMethodID  **JOIN**  Purchasing.PurchaseOrderDetail *pod* **ON** *poh*.PurchaseOrderID = *pod*.PurchaseOrderID  **WHERE**  **YEAR**(*poh*.OrderDate) = 2014  **AND** *sm*.Name **IN** ('CARGO', 'TRUCK')  **GROUP** **BY**  *sm*.Name  **ORDER** **BY**  *TotalOrderQuantity* **DESC** |
| ***Returned table*** |  |

**2.5 *(3 Marks)***

|  |  |
| --- | --- |
| ***Question*** | Calculate the sum of freight charges for each shipping method used for the orders placed in the year 2012. The result should include the shipping method and the total freight charges for that method in descending order. |
| ***Code*** | **SELECT**  *sm*.Name **AS** *ShippingMethod*,  **SUM**(*poh*.Freight) **AS** *TotalFreightCharges*  **FROM**  Purchasing.PurchaseOrderHeader *poh*  **JOIN**  Purchasing.ShipMethod *sm* **ON** *poh*.ShipMethodID = *sm*.ShipMethodID  **WHERE**  **YEAR**(*poh*.OrderDate) = 2012  **GROUP** **BY**  *sm*.Name  **ORDER** **BY**  *TotalFreightCharges* **DESC** |
| ***Returned table*** |  |

**2.6 *(3 Marks)***

|  |  |
| --- | --- |
| ***Question*** | Select the top 5 vendors with the highest total sales (calculated by multiplying the order quantity and unit price) and include their state. Display the vendor’s name, state, and total sales, and sort the results in ascending order of total sales. |
| ***Code*** | **SELECT** TOP 5  v.Name **AS** VendorName,  *vwa*.StateProvinceName **AS** *State*,  **SUM**(*pod*.OrderQty \* *pod*.UnitPrice) **AS** *TotalSales*  **FROM**  Purchasing.Vendor *v*  **JOIN**  Purchasing.vVendorWithAddresses *vwa* **ON** *v*.BusinessEntityID = *vwa*.BusinessEntityID  **JOIN**  Purchasing.PurchaseOrderHeader *poh* **ON** *v*.BusinessEntityID = *poh*.VendorID  **JOIN**  Purchasing.PurchaseOrderDetail *pod* **ON** *poh*.PurchaseOrderID = *pod*.PurchaseOrderID  **GROUP** **BY**  *v*.Name, *vwa*.StateProvinceName  **ORDER** **BY**  *TotalSales* **ASC** |
| ***Returned table*** |  |

**2.7 *(4 Marks)***

|  |  |
| --- | --- |
| ***Question*** | The company plans to stop purchasing from vendors who have an **overall** **average lead** time of over 20 days. Write a query to identify products that will have no available vendors if purchases from these poorly rated vendors are stopped. The query should return product IDs if any exist. |
| ***Code*** | **WITH** *PoorlyRatedVendors* **AS** (  **SELECT** **DISTINCT** BusinessEntityID  **FROM** Purchasing.ProductVendor  **GROUP** **BY** BusinessEntityID  **HAVING** **AVG**(AverageLeadTime) > 20  )  **SELECT** **DISTINCT** *pv*.ProductID  **FROM** Purchasing.ProductVendor *pv*  **WHERE** **NOT** **EXISTS** (  **SELECT** 1  **FROM** Purchasing.ProductVendor *pv2*  **WHERE** *pv2*.ProductID = *pv*.ProductID  **AND** *pv2*.BusinessEntityID **NOT** **IN** (**SELECT** BusinessEntityID **FROM** *PoorlyRatedVendors*)  ) |
| ***Returned table*** |  |

**2.8 *(4 Marks)***

|  |  |
| --- | --- |
| ***Question*** | The company has noticed that some vendors have an unusually high rejection rate for their supplied products. Calculate the percentage of rejected products against their total ordered quantities for each vendor and include those exceeding 5%.  Write a query to display the vendor’s name, total ordered quantity, total rejected quantity, and rejection percentage. Discuss your observations and insights along with potential actions the company can take based on the results.  *Hint: You may consider rejectedQty and OrderQty in answering the question.* |
| ***Code/s*** | **SELECT**  *v*.Name **AS** *VendorName*,  **SUM**(*pod*.OrderQty) **AS** *TotalOrderedQuantity*,  **SUM**(*pod*.RejectedQty) **AS** *TotalRejectedQuantity*,  (**SUM**(*pod*.RejectedQty) \* 100.0 / **SUM**(*pod*.OrderQty)) **AS** *RejectionPercentage*  **FROM**  Purchasing.Vendor *v*  **JOIN**  Purchasing.PurchaseOrderHeader *poh* **ON** *v*.BusinessEntityID = *poh*.VendorID  **JOIN**  Purchasing.PurchaseOrderDetail *pod* **ON** *poh*.PurchaseOrderID = *pod*.PurchaseOrderID  **GROUP** **BY**  *v*.Name  **HAVING**  (**SUM**(*pod*.RejectedQty) \* 100.0 / **SUM**(*pod*.OrderQty)) > 5  **ORDER** **BY**  *RejectionPercentage* **DESC** |
| ***Returned table/s*** | A screenshot of a computer  Description automatically generated |
| ***Discussion*** | Insights:   * Total rejected quantity from International vendor is 9 over 153 ordered product. Which shows overall quality of their products. * Premier Sport has highest amount of ordered Quantity but it also has highest amount of Rejected Quantity. This combination could be problematic due to large amount of returned product.   Potential Action:   * Quality Control Audits: These vendors should conduct quality control audits to improve quality of their products. * Performance Metrics: Establish clear performance metrics and targets for product quality, including maximum acceptable rejection rates. * Root Cause Analysis: Conduct root cause research and come up with effective strategies to tackle return product issue. * Put Target on performance key index on product evaluation. |

**2.9 *(5 Marks)***

|  |  |
| --- | --- |
| ***Question*** | The company wants to categorize its vendors into "premium", "standard" or “poor” based on relevant performance values such as credit ratings and the percentage of rejected product count. Vendors who categorised as “poor” with low credit ratings and high reject rates should be flagged for potential contract review, while vendors with strong credit ratings and low reject rates should be prioritized for long-term relationships.  Propose on criteria for categorizing vendors into "premium", "standard" and “poor” categories based on selected values such as credit ratings, percentage of rejected quantity and other performance related data. You are required to propose a criterion/criteria with justification, and write code to indicate whether vendor belongs to "premium", "standard" or “poor” category based on your criteria. |
| ***Criteria*** | Based on credit rating  High: 1-2  Medium: 3-4  Low: 5  Rejection Rate:  Low: 0-2%  Medium: 2-5%  High: >5%  Categorization:  Premium: High credit rating AND Low rejection rate  Poor: Low credit rating OR High rejection rate  Standard: All other combinations |
| ***Justification*** | * Credit rating directly reflects the vendor's financial stability and reliability. * Rejection rate will show quality of supplied product by vendor. * Combining these factors gives a view of the vendor's overall performance |
| ***Code/s*** | **WITH** *VendorRejectionRates* **AS** (  **SELECT**  *poh*.VendorID,  (**SUM**(*pod*.RejectedQty) \* 100.0 / **SUM**(*pod*.OrderQty)) **AS** *RejectionRate*  **FROM**  Purchasing.PurchaseOrderHeader *poh*  **JOIN**  Purchasing.PurchaseOrderDetail *pod* **ON** *poh*.PurchaseOrderID = *pod*.PurchaseOrderID  **GROUP** **BY**  *poh*.VendorID  )  **SELECT**  *v*.BusinessEntityID,  *v*.Name **AS** *VendorName*,  *v*.CreditRating,  **COALESCE**(*vr*.*RejectionRate*, 0) **AS** *RejectionRate*,    **CASE**  **WHEN** *v*.CreditRating <= 2 **AND** **COALESCE**(*vr*.*RejectionRate*, 0) <= 2 **THEN** 'Premium'  **WHEN** *v*.CreditRating >= 5 **OR** **COALESCE**(*vr*.*RejectionRate*, 0) > 5 **THEN** 'Poor'  **ELSE** 'Standard'  **END** **AS** *VendorCategory*  **FROM**  Purchasing.Vendor *v*  **LEFT** **JOIN**  *VendorRejectionRates* *vr* **ON** *v*.BusinessEntityID = *vr*.VendorID  **ORDER** **BY**  *VendorCategory*, *v*.Name |
| ***Returned table/s*** |  |

**2.10 *(5 Marks)***

|  |  |
| --- | --- |
| ***Question*** | The company management wants to analyse sales performance across different provinces to identify high-performing provinces and understand their sales growth trends over the years. Develop criteria to evaluate provinces based on total sales revenue and year-over-year sales growth. Write SQL queries extract required data based on your analysis.  Based on these extracted data, discuss your observations about sales trends and identify potential areas for strategic improvement. If required, you can propose criteria for categorizing provinces into "high-priority," "medium-priority," and "low-priority" categories to support your analysis. |
| ***Criteria*** | These queries give us TotalSalesRevenue and YearlyGrowth. So the purposed criteria is,   * High Priority: In each province if the total sales is above avg revenue AND positive year overgrowth. * Medium Priority: Average total sales revenue AND stable or slightly positive growth. * Low Priority: Below Average total sales revenue AND negative growth. |
| ***Justification*** | These queries give us TotalSalesRevenue and YearlyGrowth. |
| ***Code*** | **SELECT**  *vwa*.StateProvinceName **AS** *Province*,  **YEAR**(*poh*.OrderDate) **AS** *Year*,  **SUM**(*poh*.TotalDue) **AS** *TotalSalesRevenue*  **FROM**  Purchasing.PurchaseOrderHeader *poh*  **JOIN**  Purchasing.vVendorWithAddresses *vwa* **ON** *poh*.VendorID = *vwa*.BusinessEntityID  **GROUP** **BY**  *vwa*.StateProvinceName, **YEAR**(*poh*.OrderDate)  **ORDER** **BY**  *vwa*.StateProvinceName, **YEAR**(*poh*.OrderDate);  **SELECT**  *Province*,  *Year*,  *TotalSalesRevenue*,  LAG(*TotalSalesRevenue*) **OVER** (**PARTITION** **BY** *Province* **ORDER** **BY** *Year*) **AS** *PreviousYearSales*,  (*TotalSalesRevenue* - LAG(*TotalSalesRevenue*) **OVER** (**PARTITION** **BY** *Province* **ORDER** **BY** *Year*)) \* 100.0 /  LAG(*TotalSalesRevenue*) **OVER** (**PARTITION** **BY** *Province* **ORDER** **BY** *Year*) **AS** *YearOverYearGrowth*  **FROM** (  **SELECT**  *vwa*.StateProvinceName **AS** *Province*,  **YEAR**(*poh*.OrderDate) **AS** *Year*,  **SUM**(*poh*.TotalDue) **AS** *TotalSalesRevenue*  **FROM**  Purchasing.PurchaseOrderHeader *poh*  **JOIN**  Purchasing.vVendorWithAddresses *vwa* **ON** *poh*.VendorID = *vwa*.BusinessEntityID  **GROUP** **BY**  *vwa*.StateProvinceName, **YEAR**(*poh*.OrderDate)  ) **AS** *SalesData*  **ORDER** **BY**  *Province*, *Year*; |
| ***Returned table*** |  |

***NOTE:***

* *You can make assumptions when you are writing queries for questions 2.8, 2.9 and 2.10. When you are working in real industry environments, you will need to come up with criteria, assumptions and justifications, as everything will not be perfectly documented.*
* ***CreditRating:*** *1 = Superior, 2 = Excellent, 3 = Above average, 4 = Average, 5 = Below average*

# References

More details about the contents of the tables can be found under below links.

* <https://dataedo.com/samples/html/AdventureWorks/doc/AdventureWorks_2/tables/Purchasing_ProductVendor_173.html>
* <https://dataedo.com/samples/html/AdventureWorks/doc/AdventureWorks_2/tables/Purchasing_PurchaseOrderDetail_174.html>
* <https://dataedo.com/samples/html/AdventureWorks/doc/AdventureWorks_2/tables/Purchasing_PurchaseOrderHeader_175.html>
* <https://dataedo.com/samples/html/AdventureWorks/doc/AdventureWorks_2/tables/Purchasing_ShipMethod_176.html>
* <https://dataedo.com/samples/html/AdventureWorks/doc/AdventureWorks_2/tables/Purchasing_Vendor_177.html>

# 

# Submission Guide

Your submission is this WORD file with all the questions answered. The SQL code has to be copied as text (not an image or screenshot) so that the markers can immediately re-run and check your answers. Marks will not be given if the code is not provided in plain text.

Submissions with high similarity in the SQL code with another submission, especially having the same wrong code, will be considered plagiarism/collusion and will be reported to the Academic Integrity Advisors (AIA) for further investigation.

# Marking rubric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criterion | High Distinction  (> 80%) | Distinction  (70-79%) | Credit  (60-69%) | Pass  (50-59%) |
| ERD, Relational Schema, and SQL Statements (20) | Exceptional effort to address questions and present information and insights.  In depth  knowledge of ERD and database schema.  All entities and attributes are identified. | Excellent effort to address questions and present information and insights.  Comprehensive  knowledge and understanding of ERD and database schema.  Majority of entities and attributes are identified. | Fair effort to address questions and present information and insights.  Developed knowledge and understanding of ERD and database schema.  Almost majority of entities and attributes are identified. | Limited effort to address questions.  Acceptable knowledge and understanding of ERD and database schema.  Acceptable number of entities and attributes are identified. |
| Anomaly (10) | Demonstrates in depth  knowledge of anomalies. | Demonstrates comprehensive  knowledge and understanding  of anomalies. | Demonstrates a  developed knowledge  and understanding of anomalies. | Demonstrates  acceptable knowledge  and understanding of anomalies. |
| SQL code (30) | Demonstrates in depth  knowledge of constructing SQL. | Demonstrates comprehensive  knowledge and understanding  of constructing SQL. | Demonstrates a  developed knowledge  and understanding of constructing SQL. | Demonstrates  acceptable knowledge  and understanding of constructing SQL. |