|  |
| --- |
|  |
| **Task:**  Using the simplified partial given ERD (appendix B), analyse the requirements (appendix A) and then:   1. Produce a script suitable for SQL Developer to create the tables in the ERD (appendix B). Remember to define a primary and foreign key(s) where necessary using constraints. Entity definitions are found in appendix C.   You will require some numeric data types for certain attributes so that you can comply with the requirement for numeric functions in queries. Examine the data records supplied carefully to identify the numeric attributes. There is an example in appendix C.  Using this document, place a copy of your final script (the DDL) here.  CREATE TABLE invoices (  invoice\_number VARCHAR(20) PRIMARY KEY,  idate DATE,  vat\_total NUMBER,  total\_value NUMBER,  supplier\_number VARCHAR(10),  FOREIGN KEY (supplier\_number) REFERENCES supplier(supplier\_number)  );  CREATE TABLE invoice\_details (  invoice\_number VARCHAR(20),  part\_number\_with\_prefix CHAR(12),  quantity NUMBER,  net\_price NUMBER,  net\_vat NUMBER,  PRIMARY KEY (invoice\_number, part\_number\_with\_prefix),  FOREIGN KEY (invoice\_number) REFERENCES invoice(invoice\_number),  FOREIGN KEY (part\_number\_with\_prefix) REFERENCES part(part\_number\_with\_prefix)  );  CREATE TABLE locations (  part\_number\_with\_prefix CHAR(12),  bin\_number VARCHAR(10),  quantity NUMBER,  warehouse\_id VARCHAR(50),  PRIMARY KEY (part\_number\_with\_prefix, bin\_number, warehouse\_id),  FOREIGN KEY (part\_number\_with\_prefix) REFERENCES part(part\_number\_with\_prefix),  FOREIGN KEY (bin\_number, warehouse\_id) REFERENCES bin(bin\_number, warehouse\_id)  );  CREATE TABLE suppliers (  supplier\_number VARCHAR(10) PRIMARY KEY,  supplier\_name VARCHAR(100) NOT NULL,  address VARCHAR(100),  town VARCHAR(50),  county VARCHAR(50),  postcode VARCHAR(10),  contact\_number VARCHAR(20),  email\_address VARCHAR(100),  country VARCHAR(50)  );  CREATE TABLE warehouses (  warehouse\_id VARCHAR(50) PRIMARY KEY,  address VARCHAR(100),  town VARCHAR(50),  county VARCHAR(50),  postcode VARCHAR(10),  contact\_number VARCHAR(20)  );  CREATE TABLE bins (  bin\_number VARCHAR(10),  warehouse\_id VARCHAR(50),  PRIMARY KEY (bin\_number, warehouse\_id),  FOREIGN KEY (warehouse\_id) REFERENCES warehouse(warehouse\_id)  );  CREATE TABLE parts (  part\_number\_with\_prefix CHAR(12) PRIMARY KEY,  part\_number CHAR(10) NOT NULL,  description VARCHAR(40),  discount\_sw\_code CHAR(4),  discount\_sw\_percent NUMBER(3),  krona\_net NUMBER,  discount\_uk\_code CHAR(4),  discount\_uk\_percent NUMBER(3),  sterling\_net NUMBER  );    Populate the tables with the .sql files supplied in the folder ‘SQL Scripts’ on BlackBoard. Note, these files must be opened in Notepad (Windows) or TextEdit (macOS) from where you can then copy/paste them into an SQL worksheet.   1. Write SQL statements necessary to satisfy the following queries on page 3. Remember to use sensible headings where appropriate.   The queries will retrieve data from single and multiple tables, they will use textual and numeric functions as well as creating views.  Again, using this document, place a copy of the DML (SQL for creation of queries) together with the results or outputs of these queries under each question below.  Please ensure that the output columns are ‘lined up’ and nicely formatted when copying and pasting from the SQL interface.  **Queries:**   1. Produce a ‘telephone list’ for all suppliers. Define a ‘View’ and include the supplier name and telephone number. Sort by supplier ascending. The final output should only be one column.   **(5 marks)**  CREATE VIEW Suppliers\_Phonebook AS  SELECT supplier\_name, contact\_number  FROM suppliers  ORDER BY supplier\_name;  select contact\_number from Suppliers\_Phonebook;     1. Create a query to establish which invoices are over £200. Sort on TOTAL\_VALUE descending.   **(3 marks)**  SELECT \*  FROM invoice  WHERE total\_value > 200  ORDER BY total\_value DESC;     1. Create a query to select a particular invoice of your choice. The user should be prompted for the criteria – i.e. INVOICE\_NUMBER should be entered at run-time.   Select…From…Where INVOICE\_NUMBER = 100001 is INCORRECT  **(4 marks)**  SELECT \*  FROM invoices  WHERE invoice\_number = &invoice\_number;       1. Produce a list of parts for both warehouses. Include BIN\_NUMBER and QUANTITY as well as other suitable attributes. This should be one query. Sort by WAREHOUSE\_ID. Use correlation (alias) names.   **(4 marks)**  SELECT w.warehouse\_id, l.quantity, b.bin\_number  FROM warehouses w  INNER JOIN bins b ON b.warehouse\_id = w.warehouse\_id  INNER JOIN locations l ON l.bin\_number = b.bin\_number  ORDER BY w.warehouse\_id;     1. Produce a list of parts to find the value of stock by line (Qty x Unit Price) in the Haverfordwest warehouse. Order by PART\_NUMBER\_WITH\_PREFIX.   **(4 marks)**  SELECT  p.part\_number\_with\_prefix,  l.quantity,  p.sterling\_net,  (l.quantity \* p.sterling\_net) AS line\_value  FROM  parts p  INNER JOIN locations l ON l.part\_number\_with\_prefix = p.part\_number\_with\_prefix  INNER JOIN bins b ON b.bin\_number = l.bin\_number  INNER JOIN warehouses w ON w.warehouse\_id = b.warehouse\_id  WHERE  w.warehouse\_id = 'Steve Haverfordwest'  ORDER BY  p.part\_number\_with\_prefix;     1. Create a View from your query 5 and then use it to calculate the total value of all stock currently held in the Haverfordwest warehouse. Sort on PART\_NUMBER then create another view to calulate the total stock value.   **(6 marks)**  CREATE VIEW LineValues AS  SELECT  p.part\_number\_with\_prefix,  l.quantity,  p.sterling\_net,  (l.quantity \* p.sterling\_net) AS line\_value  FROM  part p  INNER JOIN locations l ON l.part\_number\_with\_prefix = p.part\_number\_with\_prefix  INNER JOIN bins b ON b.bin\_number = l.bin\_number  INNER JOIN warehouses w ON w.warehouse\_id = b.warehouse\_id  WHERE  w.warehouse\_id = 'Steve Haverfordwest'  ORDER BY  p.part\_number\_with\_prefix;  CREATE VIEW TotalStockValue AS  SELECT  SUM(line\_value) AS total\_value  FROM  LineValues;  **select \* from LineValues**    **select \* from TotalStockValue**   1. Produce a list of parts that are out of stock in both warehouses.   **(4 marks)**  SELECT p.part\_number\_with\_prefix  FROM parts p  LEFT JOIN locations l1 ON l1.part\_number\_with\_prefix = p.part\_number\_with\_prefix AND l1.warehouse\_id = 'Steve Haverfordwest'  LEFT JOIN locations l2 ON l2.part\_number\_with\_prefix = p.part\_number\_with\_prefix AND l2.warehouse\_id = 'Peter Essex'  WHERE  l1.quantity IS NULL  AND l2.quantity IS NULL;     1. Produce a list of parts that are available from both warehouses. You will need two queries and a view.   **(10 marks)**  CREATE VIEW haverfordwest AS  SELECT part\_number\_with\_prefix  FROM locations  WHERE warehouse\_id = 'Steve Haverfordwest';  CREATE VIEW essex AS  SELECT part\_number\_with\_prefix  FROM locations  WHERE warehouse\_id = 'Peter Essex';  CREATE VIEW CombinedWarehouses AS  SELECT part\_number\_with\_prefix  FROM haverfordwest  INTERSECT  SELECT part\_number\_with\_prefix  FROM essex;  select \* from CombinedWarehouses;       1. Create a query to find out which parts are cheaper to buy from Sweden, when converted from Swedish Krona to Sterling and rounded to 2 decimal places. Use PART\_NUMBER\_WITH\_PREFIX, STERLING\_NET and your third calculated column. To convert krona to sterling use 13/100.   **(10 marks)**  SELECT  part\_number\_with\_prefix,  sterling\_net,  ROUND(krona\_net \* 0.13, 2) AS krona\_converted,  CASE  WHEN krona\_net \* 0.13 < sterling\_net THEN 'Sweden'  ELSE 'UK'  END AS cheaper\_from  FROM parts;     1. Produce a query to increase the cost of all parts by 5%. Use PART\_NUMBER\_WITH\_PREFIX, STERLING\_NET and your new price including the 5%. Order by the new price.   **(3 marks)**  SELECT  part\_number\_with\_prefix,  sterling\_net,  ROUND(sterling\_net \* 1.05, 2) AS new\_price  FROM parts  ORDER BY new\_price; |

**Learning Outcomes to be assessed** (as specified in the validated module descriptor <http://icis.glam.ac.uk>):

To demonstrate a practical understanding of the design of information systems.

|  |  |  |
| --- | --- | --- |
| **Marking Scheme** | **Marks Available** | **Marks Awarded** |
| Table definitions in SQL (DDL) | 35 |  |
| Specified SQL statements (DML) | 55 |  |
| Exceptional elegance of SQL queries | 10 |  |
| **Total** | 100 |  |

**Assessment Criteria:**

Please see Appendix D

**Appendix A**

EuroParts.com is an established company in Wales. It deals with the import and export of car parts. The company’s suppliers are based in the UK and Sweden.

EuroParts.com has a warehouse in Haverfordwest and another in Stansted and both hold a stock of parts. Steve is manager of the Haverfordwest branch and Peter manages the Stansted warehouse. Each warehouse has many storage bins comprised of shelves and each shelf will have many plastic part storage containers which are numbered. For simplicity only use the shelf numbers.

Part prices are in two currencies, UK sterling, and Swedish krona. All part prices are net of Vat.

1. Each warehouse would have a unique set of bins numbers i.e., bin number A1M located in Warehouse ID “Steve Haverfordwest” would not be found in Warehouse ID “Peter Essex”.
2. Some bin shelves (location) may be empty.
3. There will be no requirement for additional warehouses.
4. There may be new suppliers.
5. There may be new parts.
6. The same part may be kept in both warehouses.
7. The entity “Location” is the shelf location on a bin. A bin will have several shelves.
8. The part number ‘prefix’, the first two digits, identifies whether the part is for bodywork, engine, accessory etc.
9. The discount code and discount percentage are the markup on the part (e.g., Discount\_UK\_Code of 1 has a Discount\_UK\_Percent of 25 which means 25% profit). The net cost of parts is in both UK sterling and Swedish Krona.
10. For simplicity the Invoice contains a minimum number of attributes.

There is a M:N relationship between supplier and part but this will not be investigated for the purposes of this assignment. It can be ignored. There would also be a relationship between supplier and warehouse but this is also outside the remit of this coursework.

# Appendix B

# Entity-Relationship Diagram

**A simplified and partial ERD for EuroParts.com**

1

accommodates

is placed in a

contains

sends

is composed of

1

0..\*

1

1

1

1..\*

1

0..\*

Bin

Warehouse

Location

Invoice detail

Invoice

Part

Supplier

0..\*

These are LINK entities.

## Appendix C

holds

1..\*

1..\*

## Entity Definitions/Attributes:

**Part** (part\_number\_with\_prefix, part\_number, description, discount\_sw\_code, discount\_sw\_percent, krona\_net, discount\_uk\_code, discount\_uk\_percent, sterling net)

**Supplier** (supplier\_number, supplier\_name, address, town, county, postcode, contact\_number, email\_address, country)

**Warehouse** (warehouse\_id, address, town, county, postcode, contact\_number)

**Bin** (bin\_number, warehouse\_id)

**Location** (part\_number\_with\_prefix, bin\_number, quantity)

**Invoice** (invoice\_number, idate, vat\_total, total\_value, supplier\_number)

**InvoiceDetail** (invoice\_number, part\_number\_with\_prefix, quantity, net\_price, net\_vat)

**Use part\_number\_with\_prefix as the PK for the part table.**

An example for data types for ‘Part’ is shown below:

**Part**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| PART\_NUMBER\_WITH\_PREFIX | NOT NULL | CHAR(12) |
| PART\_NUMBER |  | CHAR(10) |
| DESCRIPTION |  | VARCHAR2(40) |
| DISCOUNT\_SW\_CODE |  | CHAR(4) |
| DISCOUNT\_SW\_PERCENT |  | NUMBER(3) |
| KRONA\_NET |  | NUMBER |
| DISCOUNT\_UK\_CODE |  | CHAR(4) |
| DISCOUNT\_UK\_PERCENT |  | NUMBER(3) |
| STERLING\_NET |  | NUMBER |

# Populate tables with the following data:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PART\_NUMBER\_WITH\_PREFIX**  **PART** | **PART\_NUMBER** | **DESCRIPTION** | **DISCOUNT\_SW\_CODE** | **DISCOUNT\_SW\_PERCENT** | **KRONA\_NET** | **DISCOUNT\_UK\_CODE** | **DISCOUNT\_UK\_PERCENT** | **STERLING\_NET** |
| 104531331 | 4531331 | Bushing | D51 | 35 | 38.81 | 1 | 25 | 3.68 |
| 105112495 | 5112495 | Retaining Ring | D51 | 35 | 5.48 | 1 | 25 | .49 |
| 104543518 | 4543518 | Bushing | D51 | 35 | 25.44 | 1 | 25 | 2.19 |
| 104246112 | 4246112 | Rubber Block | D51 | 35 | 5.44 | 1 | 25 | .74 |
| 1055557379 | 55557379 | Chain Gear | D01 | 30 | 177.34 | 90 | 45 | 12.62 |
| 109178336 | 9178336 | Chain Gear | D01 | 30 | 177.05 | 90 | 45 | 12.29 |
| 105450192 | 5450192 | Chain Gear | D01 | 30 | 519.98 | 1 | 25 | 51.68 |
| 104411997 | 4411997 | Switch | D51 | 35 | 116.47 | 1 | 25 | 11.59 |
| 107585086 | 7585086 | Chain Tension | D51 | 35 | 536.55 | 90 | 45 | 43.93 |
| 108373078 | 8373078 | bearing | D51 | 35 | 16.96 | 90 | 45 | 1.93 |
| 109129669 | 9129669 | Cylinder Head | C00 | 18 | 12160.17 | 2 | 10 | 1099.87 |
| 109522822 | 9522822 | Relay | D51 | 35 | 377.67 | 1 | 25 | 44.39 |
| 104161162 | 4161162 | Seals | D01 | 30 | 15.53 | 1 | 25 | 1.42 |
| 107522733 | 7522733 | Fuel Pump | D81 | 40 | 72.36 | 36 | 50 | 29.33 |
| 107536923 | 7536923 | Fuel Pump | D81 | 40 | 1331.47 | 36 | 50 | 201.41 |
| 105954557 | 5954557 | Tooth Belt | D81 | 40 | 1116.06 | 90 | 45 | 74.59 |
| 1055556404 | 55556404 | Belt | D81 | 40 | 64.86 | 36 | 50 | 5.11 |
| 1093185051 | 93185051 | Belt | D81 | 40 | 128.5 | 36 | 50 | 19.16 |
| 1093185049 | 93185049 | Belt | D81 | 40 | 92.84 | 36 | 50 | 16.78 |
| 1093185050 | 93185050 | Belt | D81 | 40 | 175.5 | 36 | 50 | 16.62 |
| 1012795070 | 12795070 | Belt | D51 | 35 | 12.22 | 90 | 45 | 10.22 |
| 10930600 | 930600 | Seal | D51 | 35 | 41.42 | 90 | 45 | 22.11 |
| 104109088 | 4109088 | Gear | D51 | 35 | 99.45 | 90 | 45 | 38.99 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SUPPLIER\_NUMBER** | **SUPPLIER\_NAME** | **ADDRESS** | **TOWN** | **COUNTY** | **POSTCODE** | **CONTACT\_NUMBER** | **EMAIL\_ADDRESS** | **COUNTRY** |
| H001 | Higher Oak | Oak Road, Wrexham Ind Est | Wrexham | Wrexham | LL13 9RG | 01443456123 | HighOak@hotmail.co.uk | Wales |
| N001 | Nordic Car Company | Unit 2, ByFleet Technical Centre | ByFleet | Surrey | KT14 7JL | 01798445566 | NordicCars@tiscali.co.uk | Great Britain |
| S002 | Swain and Jones | 35-42 East Street | Farnham | Surrey | GU9 7SW | 01798776589 | French1@hotmail.co.uk | Wales |
| U001 | Ultimate Car Force | Unit 4, Burrows Ind Est | Shere, Guildford | Surrey | GU55 9QQ | 01567836425 | UltimateCars@yahoo.com | Great Britain |
| W001 | Workshop Saab Specialists | 9 Clothier Road | Brislington | Bristol | BS4 5PS | 01792623594 | TheWorkshop@btconnect.com | Great Britain |
| A001 | Abbot Racing | Spinnels Farm, Wix | Manningtree | Essex | CO11 2UJ | 01279641225 | AbbottRacing@yahoo.co.uk | Great Britain |
| A002 | Autohaus Furst GMBH | Berbillger Str 4 | 71254 Ditzingen |  |  | 004971191893 | Autohaus@yahoo.de | Germany |
| B001 | Bond Street | Kerry House, 108 Vaughan Way | Leicester | Leicester | LE2 6HJ | 01664765533 | BondStreetSaab@hotmail.co.uk | Great Britain |
| O001 | Otto Olssons Bil AB | Box 94 | 27322 | Tomelilla |  | 004641778100 | OttoOlsson@hotmail.se | Sweden |
| S003 | Saab Owners Club | 16 Thistle Nest Close | Otley | West Yorkshire | LS21 2RR | 01552786632 | SaabOwners@hotmail.com | Great Britain |
| F001 | French and Swedish Car Centre | 32 Merlins Bridge | Haverfordwest | Pembrokeshire | SA61 5SZ | 01437766511 | FrenchSwedish@hotmail.com | Wales |
| M001 | Merlin Auto Centre | Merlins Bridge | Haverfordwest | Pembrokeshire | SA61 4GH | 01437765522 | MerlinAutoCentre@hotmail.com | Wales |
| S001 | Swedish Parts Centre | Unit 4, Chatterly Industrial Estate | Stansted | Essex | CM23 5HJ | 01279563214 | SwedishParts@btconnect.com | Great Britain |

**SUPPLIER**

**WAREHOUSE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **WAREHOUSE\_ID** | **ADDRESS** | **TOWN** | **COUNTY** | **POSTCODE** | **CONTACT\_NUMBER** |
| Steve Haverfordwest | Riverview, Spittal | Haverfordwest | Pembrokeshire | SA62 3BA | 01437 123456 |
| Peter Essex | Little Mays | Dunmow | Essex | CM23 5TT | 01371 870000 |

|  |  |
| --- | --- |
| **BIN\_NUMBER** | **WAREHOUSE\_ID** |
| O137 | Peter Essex |
| O134 | Peter Essex |
| O138 | Peter Essex |
| O133 | Peter Essex |
| O129 | Peter Essex |
| O128 | Peter Essex |
| O127 | Peter Essex |
| O126 | Peter Essex |
| O124 | Peter Essex |
| O121 | Peter Essex |
| O120 | Peter Essex |
| O119 | Peter Essex |
| O118 | Peter Essex |
| O117 | Peter Essex |
| O114 | Peter Essex |
| O113 | Peter Essex |
| O112 | Peter Essex |
| N517 | Peter Essex |
| N516 | Peter Essex |
| N515 | Peter Essex |
| N514 | Peter Essex |
| N513 | Peter Essex |
| N512 | Peter Essex |
| N511 | Peter Essex |

**BIN**

|  |  |
| --- | --- |
| **BIN\_NUMBER** | **WAREHOUSE\_ID** |
| A2L | Steve Haverfordwest |
| A2T | Steve Haverfordwest |
| A2M | Steve Haverfordwest |
| A1L | Steve Haverfordwest |
| A1M | Steve Haverfordwest |
| A1T | Steve Haverfordwest |
| A15M | Steve Haverfordwest |
| A15T | Steve Haverfordwest |
| A15L | Steve Haverfordwest |
| A9L | Steve Haverfordwest |
| A9M | Steve Haverfordwest |
| A9T | Steve Haverfordwest |
| A4L | Steve Haverfordwest |
| A4M | Steve Haverfordwest |
| A4T | Steve Haverfordwest |
| A5T | Steve Haverfordwest |
| A5M | Steve Haverfordwest |
| A5L | Steve Haverfordwest |
| A6L | Steve Haverfordwest |
| A6M | Steve Haverfordwest |
| A6T | Steve Haverfordwest |

**LOCATION**

|  |  |  |
| --- | --- | --- |
| **PART\_NUMBER\_WITH\_PREFIX** | **BIN\_NUMBER** | **QUANTITY** |
| 104531331 | O137 | 25 |
| 104531331 | A1M | 20 |
| 105112495 | O133 | 50 |
| 105112495 | A1T | 50 |
| 104543518 | A1L | 25 |
| 104543518 | O134 | 35 |
| 104246112 | O133 | 135 |
| 1055557379 | O129 | 5 |
| 109178336 | O128 | 2 |
| 105450192 | A2L | 12 |
| 104411997 | A2L | 15 |
| 104411997 | O126 | 0 |
| 107585086 | A5T | 0 |
| 108373078 | A5M | 0 |
| 109129669 | A9M | 1 |
| 10930600 | O120 | 1 |
| 109522822 | O119 | 9 |
| 1012795070 | O118 | 200 |
| 104109088 | O117 | 20 |
| 104161162 | A1L | 126 |
| 107522733 | A1M | 26 |
| 107536923 | A15M | 16 |

**INVOICE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **INVOICE\_NUMBER** | **IDATE** | **VAT\_TOTAL** | **TOTAL\_VALUE** | **SUPPLIER\_NUMBER** |
| 100001 | 01-NOV-23 | 1.42 | 8.1 | N001 |
| 100002 | 01-NOV-23 | 12.39 | 70.77 | S001 |
| 100003 | 02-NOV-23 | 33.49 | 144.79 | N001 |
| 100004 | 02-NOV-23 | 14.26 | 351.49 | O001 |
| 100005 | 03-NOV-23 | 192.48 | 1099.87 | O001 |
| 100006 | 03-NOV-23 | 35.25 | 201.41 | O001 |
| 100007 | 04-NOV-23 | 25.79 | 149.22 | O001 |
| 100008 | 11-OCT-23 | 1.42 | 8.1 | N001 |
| 100009 | 10-OCT-23 | 1.42 | 8.1 | N001 |
| 100010 | 11-OCT-23 | 12.39 | 70.77 | S001 |

**INVOICE DETAIL**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **INVOICE\_NUMBER** | **PART\_NUMBER\_WITH\_PREFIX** | **QUANTITY** | **NET\_PRICE** | **NET\_VAT** |
| 100001 | 104531331 | 2 | 7.36 | 1.29 |
| 100001 | 104246112 | 1 | .74 | .13 |
| 100002 | 1055557379 | 1 | 12.62 | 2.21 |
| 100002 | 109178336 | 1 | 12.29 | 2.15 |
| 100002 | 107585086 | 1 | 43.93 | 7.47 |
| 100002 | 108373078 | 2 | 1.93 | .23 |
| 100003 | 1055556404 | 2 | 10.22 | 1.79 |
| 100003 | 107522733 | 4 | 117.32 | 20.53 |
| 100003 | 1093185049 | 1 | 16.78 | 2.94 |
| 100003 | 1012795070 | 100 | 47 | 8.23 |
| 100004 | 1093185050 | 1 | 16.62 | 2.91 |
| 100004 | 109178336 | 3 | 36.87 | 6.45 |
| 100004 | 1012795070 | 10 | 4.7 | .82 |
| 100004 | 107522733 | 10 | 293.3 | 4.08 |
| 100006 | 107536923 | 1 | 201.41 | 35.25 |
| 100007 | 108373078 | 1 | 1.93 | .23 |
| 100007 | 107585086 | 1 | 43.93 | 7.47 |
| 100007 | 105450192 | 2 | 103.36 | 18.09 |
| 100005 | 109129669 | 1 | 1099.87 | 192.48 |
| 100008 | 104531331 | 2 | 7.36 | 1.29 |
| 100008 | 104246112 | 1 | .74 | .13 |
| 100009 | 104531331 | 2 | 7.36 | 1.29 |
| 100009 | 104246112 | 1 | .74 | .13 |
| 100010 | 1055557379 | 1 | 12.62 | 2.21 |
| 100010 | 109178336 | 1 | 12.29 | 2.15 |
| 100010 | 107585086 | 1 | 43.93 | 7.47 |
| 100010 | 108373078 | 2 | 1.93 | .23 |

**Appendix D**

# ASSESSMENT CRITERIA

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Fail | Narrow Fail | 3rd Class / Pass | Lower 2nd Class / Pass | Upper 2nd Class / Merit | 1st Class / Distinction |
| Table Definitions in SQL 35% | * Relations/Tables missing. Primary keys, foreign keys and not null/null used incorrectly in most instances | * Relations/Tables missing. Primary keys, foreign keys and not null/null used incorrectly in most instances | * No Relations/Tables missing. Primary keys, foreign keys and not null/null used incorrectly in many instances | * No Relations/Tables missing. Primary keys, foreign keys and not null/null used incorrectly in some instances | * No Relations/Tables missing. Primary keys, foreign keys and not null/null used correctly in many instances | * No Relations/Tables missing. Primary keys, foreign keys and not null/null used correctly in most/all instances |
| SQL DML statements 55% | * Very few SQL statements that meet the information retrieval requirements detailed in the scenario | * Many SQL statements that do not meet the information retrieval requirements detailed in the scenario | * SQL statements that meet some of the information retrieval requirements detailed in the scenario | * Some SQL statements that do not meet the information retrieval requirements detailed in the scenario | * SQL statements that meet many of the information retrieval requirements detailed in the scenario | * SQL statements that meet all of the information retrieval requirements detailed in the scenario |
| Exceptional elegance of SQL queries 10% | * No attempt at SQL statements that produce innovative/ exceptional code | * V. Little attempt at SQL statements that produce innovative/ exceptional code | * Little attempt at SQL statements that produce innovative/ exceptional code | * Some attempt at SQL statements that produce innovative/ exceptional code | * Some very good attempts at SQL statements that produce innovative/ exceptional code | * Excellent attempt at SQL statements that produce innovative/ exceptional code |