

Task	Expected response	Additional guidance	Marks available	
2	Database design and development			
2a	<p>Any two of the following for 1 mark each:</p> <ul style="list-style-type: none"> ♦ A query (using SUM) to add up customer spend ♦ A query (using a calculation) to find how many vouchers have been sold by a single supplier ♦ A query to search for offers within a specific category ♦ A query to sort offers from best to worst ♦ Queries to update, insert or delete data 	<p>Functional requirements should be extracted from end-user information.</p> <p>Award a maximum of 1 mark for update, insert or delete example</p>	2	Analysis
2b	<p>1 mark each for:</p> <ul style="list-style-type: none"> ♦ entity names, in the correct order ♦ adding the correct number of instances ♦ adding the correct associations 	<p>Entities could be written in reverse order</p>	3	Design

Task	Expected response	Additional guidance	Marks available	
2	Database design and development			
2c	<p>1 mark each for:</p> <ul style="list-style-type: none"> ◆ Fields and alias <ul style="list-style-type: none"> ○ firstname, surname, voucherID ○ calculation as Amount of Money Spent on Voucher £ ◆ Equijoins and category criteria ◆ Wildcard 	<pre>SELECT firstname, surname, Voucher.voucherID, (quantityPurchased * price£) AS [Amount of Money Spent on Voucher £] FROM Customer, CustomerOrder, Voucher WHERE Customer.custID = CustomerOrder.custID AND Voucher.voucherID = CustomerOrder.voucherID and category = "Adventure" AND voucherName LIKE "*escape room*";</pre> <p>Accept alternatives of escape room wildcard that produce the correct output.</p>	3	
2d	<p>1 mark each for:</p> <ul style="list-style-type: none"> ◆ Sum to find number of vouchers already purchased 	<p>Combined query</p> <pre>SELECT Voucher.voucherID, supplierName, voucherName,</pre>	4	Implementatio

Task	Expected response	Additional guidance	Marks available	
2	Database design and development			
	<ul style="list-style-type: none"> ♦ Calculation to subtract quantity purchased from quantity available ♦ Correct fields, tables and aliases ♦ Equijoins and criteria 	(quantityAvailable - SUM(quantityPurchased)) AS [Still Available] FROM CustomerOrder, Voucher, Supplier WHERE CustomerOrder.voucherID = Voucher.voucherID AND Voucher.supplierCode = Supplier.supplierCode AND Voucher.voucherID = "V543" GROUP BY Voucher.voucherID, supplierName, voucherName, quantityAvailable;		
2e	1 mark each for: <ul style="list-style-type: none"> ♦ Change aggregate function to Count(*) ♦ ORDER by aggregate function descending 		2	Testing

Task	Expected response	Additional guidance	Marks available	
2	Database design and development			
2f	Any one bullet <ul style="list-style-type: none"> ♦ Expiry date might be in the past ♦ Invalid category entered ♦ Invalid number of vouchers /price 		1	Evaluation