AFRICAN LEADERSHIP UNIVERSITY
COMPUTER SCIENCE COURSE
DATABASE MANAGEMENT
RUNTIME TERROR

Questions	Queries	Results

How many orders happened in the respective years for which payments were made and confirmed?	`SELECT Quantity AS order_quantity,`
	`SUBSTR(orders.Time_stamp, 1, 4) AS`
	`year`
	`FROM orders INNER JOIN payment_amount`
	`ON orders.Order_ID =`
	`payment_amount.Order_ID`
	`WHERE payment_amount IS NOT NULL`
	`GROUP BY year`
	`ORDER BY year DESC;`
		According to the database, we managed to find that there were:
		● **44** orders in the year 2018
		● **1** order placed in the year 2019.
		NB: This were orders that were placed and confirmed by the customers.

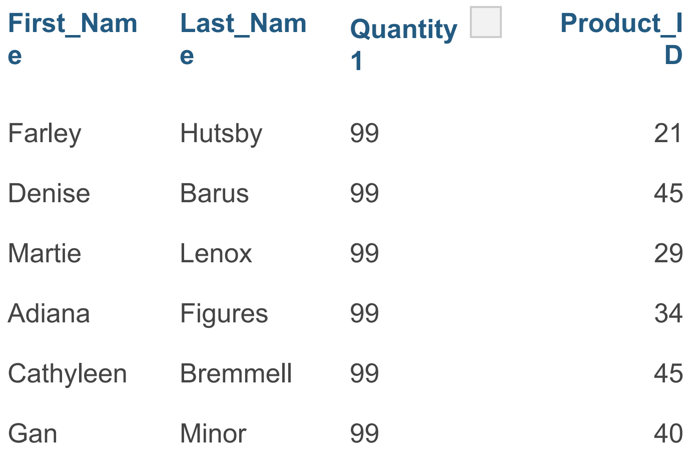
When was the first order placed by the customer on this database? What was the account of the order?	`SELECT Time_stamp, Account_Name FROM`
	`orders`
	`INNER JOIN user_account`
	`ON user_account.Account_ID =`
	`orders.Order_ID`
	`ORDER BY orders.Order_ID DESC`
	`LIMIT 1;`
		From the database, we found that the first order was placed on **14-10-2019** and the Account owner was **slaynerr**
		
		The Database result:
		
		Time_stamp **Account_Name**
		2019-10-14 **slaynerr**

Write a simple query to determine which customer/s placed the most order in 2019 and clearly articulate the Product_ID, First Name and Last Name and how many quantity/ies ordered?

```
SELECT customers.First_Name,
customers.Last_Name, Quantity,
orders.Product_ID FROM
orders INNER JOIN user_account ON
user_account.Account_ID =
orders.Account_ID
INNER JOIN customers ON
customers.Customer_ID =
user_account.Customer_ID
WHERE SUBSTR(orders.Time_stamp, 1, 4)
= '2019'
ORDER BY Quantity DESC
LIMIT 6;
```

As the database had multiple customers who had the most orders. We obtained six customers with most placed an order on the same year.

Below is the result of the query:



First_Name	Last_Name	Quantity1 □	Product_ID
Farley	Hutsby	99	21
Denise	Barus	99	45
Martie	Lenox	99	29
Adiana	Figures	99	34
Cathyleen	Bremmell	99	45
Gan	Minor	99	40

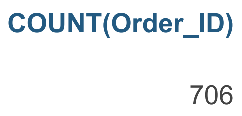
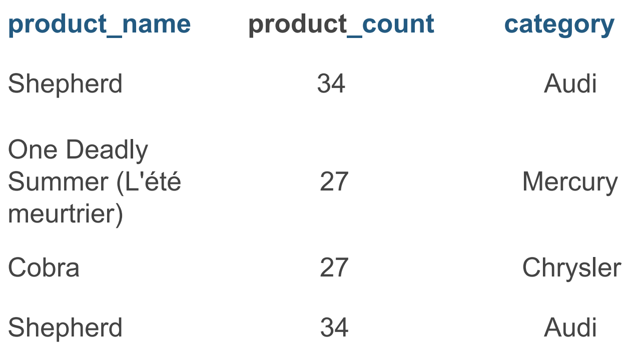
<table>
<tr><td></td><td></td><td></td></tr>
<tr><td>Develop a query to determine the top ten average payments for each product for the most recent year?</td><td>SELECT products.Product_name,
AVG(payment_amount) AS AVG_payment
FROM payment
INNER JOIN payment_amount ON
payment.Payment_amount_id =
payment_amount.Payment_amount_id
INNER JOIN orders ON orders.Order_ID
= payment.Order_ID
INNER JOIN products ON
products.Product_ID =
orders.Product_ID
WHERE SUBSTR(payment.Time_stamp, 1,
4) = '2019'
GROUP BY products.Product_name
ORDER BY AVG_payment DESC
LIMIT 10;</td><td>The Database showed the following result:<table>
<tr><th>Product_name</th><th>Avg_payment</th></tr>
<tr><td>Masters of the Universe</td><td>513.2833333333333</td></tr>
<tr><td>360</td><td>374.2288888888889</td></tr>
<tr><td>One Deadly Summer (L'été meurtrier)</td><td>362.85999999999996</td></tr>
<tr><td>Seven Sinners</td><td>353.86375</td></tr>
<tr><td>Police Academy 2: Their First Assignment</td><td>341.7207142857142</td></tr>
<tr><td>Pontypool</td><td>340.0012500000001</td></tr>
<tr><td>Reformer and the Redhead, The</td><td>314.69769230769236</td></tr>
<tr><td>Descent: Part 2, The</td><td>308.72272727272724</td></tr>
<tr><td>Lauderdale (a.k.a. Spring Break USA) (a.k.a. Sprin</td><td>297.02466666666663</td></tr>
<tr><td>Free Men (Les hommes libres)</td><td>286.36375000000004</td></tr>
</table></td></tr>
<tr><td>Calculate the minimum and maximum payments made on this database?</td><td>SELECT
min(payment_amount.payment_amount) AS
minimum_payment_amount,
max(payment_amount.payment_amount) AS
maximum_payment_amount,</td><td>The result obtained from the Database are:
● Maximum Payment being <b>996.60</b>
● Minimum payment being <b>0.16</b>
● Maximum orders being <b>1000</b></td></tr>
</table>

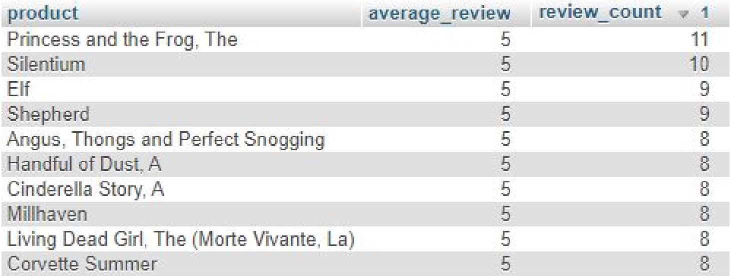
	min(orders.Order_ID) AS min_order, max(orders.Order_ID) AS max_order FROM payment_amount INNER JOIN orders ON payment_amount.Order_ID = orders.Order_ID	● Minimum orders being **1**

Calculate the total number of orders shipped and reached their desired destinations? Be sure to specify the status of the commodities being either received or not by the customer?	SELECT COUNT(Shipping_ID) AS
	Total_Number_Order
	FROM `shipping`
	WHERE Shipping_Status = 1 and
	Delivery_Status = 1;
		As the question asked, the database showed that there were 757 orders that were shipped and reached their destination as they claimed to receive the commodities by the customers.
		The result from the Database:
		Total_Number_Order
		757

Write a simple query to determine the last log-in details made in this database? Who was the customer and (s)he ordered something?	SELECT user_account.Account_ID, user_account.Account_Name, orders.Order_ID, products.Product_name FROM user_account JOIN orders ON user_account.Account_ID = orders.Account_ID JOIN products ON orders.Product_ID = products.Product_ID ORDER BY user_account.Account_ID DESC LIMIT 1;	According to the Database data, we found that the last login was done by Customer named jbuntirr with order_Id 684 and she ordered a product with a name Miss Congeniality. The result of the Database were: **Account_ID** | **Account_Name** | **Order_ID** | **Product_name** 999 | pmacfarlanrq | 921 | Pontypool

<table>
<tr><th></th><th></th><th></th></tr>
<tr>
<td>Develop a query to calculate the total number of orders placed and managed to get paid by the customer?</td>
<td><code>SELECT COUNT(payment_amount.Order_ID)</code>
<code>from payment_amount</code>
<code>WHERE payment_amount.Payment_amount</code>
<code>IS NOT NULL;</code></td>
<td>As far as the E-commerce database is concerned, it become apparent that after running the query we obtained a maximum <b>659</b> orders that were placed by the customers and got paid.

Result from the database:

COUNT(Order_ID)
706</td>
</tr>
<tr>
<td>What were the top 3 products by the number of orders placed? State the category? Include the query you run to obtain the correct result.</td>
<td><code>SELECT products.Product_name As</code>
<code>product_name,</code>
<code>COUNT(products.Product_name) AS</code>
<code>product_count, products.Category_name</code>
<code>AS category</code>
<code>FROM orders</code>
<code>JOIN products</code>
<code>ON orders.Product_ID =</code>
<code>products.Product_ID</code>
<code>GROUP BY Product_name</code>
<code>ORDER BY category_count DESC</code>
<code>LIMIT 3</code></td>
<td>From the database we obtained the top three products being:

<table>
<tr><th>product_name</th><th>product_count</th><th>category</th></tr>
<tr><td>Shepherd</td><td>34</td><td>Audi</td></tr>
<tr><td>One Deadly Summer (L'été meurtrier)</td><td>27</td><td>Mercury</td></tr>
<tr><td>Cobra</td><td>27</td><td>Chrysler</td></tr>
<tr><td>Shepherd</td><td>34</td><td>Audi</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><th></th><th></th><th></th></tr>
<tr><td>What are the top 3 cities by the number of customers registered from that city in the database?</td><td>SELECT City, count(Customer_ID)
FROM customers
GROUP BY City
ORDER BY COUNT(Customer_ID) DESC
Limit 3</td><td>From the database we obtain the top three cities with number of customers registered are:
● Stockholm
● Saguenay
● Cibodas</td></tr>
<tr><td>What are the top 10 products that recieved a 5 star rating based on the number of user who gave reviews</td><td>SELECT products.Product_name AS product, AVG(reviews.Rating) AS average_review,
COUNT(reviews.Product_ID) AS review_count
FROM reviews
JOIN products
ON reviews.Product_ID = products.Product_ID
WHERE Rating = 5
GROUP BY product
ORDER BY review_count DESC
LIMIT 10</td><td>From the database there were twenty five products that were rated 5 stars.
<table>
<tr><th>product</th><th>average_review</th><th>review_count ▼ 1</th></tr>
<tr><td>Princess and the Frog, The</td><td>5</td><td>11</td></tr>
<tr><td>Silentium</td><td>5</td><td>10</td></tr>
<tr><td>Elf</td><td>5</td><td>9</td></tr>
<tr><td>Shepherd</td><td>5</td><td>9</td></tr>
<tr><td>Angus, Thongs and Perfect Snogging</td><td>5</td><td>8</td></tr>
<tr><td>Handful of Dust, A</td><td>5</td><td>8</td></tr>
<tr><td>Cinderella Story, A</td><td>5</td><td>8</td></tr>
<tr><td>Millhaven</td><td>5</td><td>8</td></tr>
<tr><td>Living Dead Girl, The (Morte Vivante, La)</td><td>5</td><td>8</td></tr>
<tr><td>Corvette Summer</td><td>5</td><td>8</td></tr>
</table></td></tr>
</table>