DQL Tasks

-- 1) How many orders were received for products with a category\_id = 2

-- 2) How many orders were received with a category\_id of either 2, 4, or 5

-- 3) How many order are there with a price over £35.00

-- 4) How many orders are there where the customer has a date of birth before 1st January 1980 and want to receive the newsletter

-- 5) How many customers named Davenport placed orders?

-- 6) Which customer with a firstname starting with 'Br', had the most orders

-- 7) List all orders with products from category 3 by order of price, highest first.

-- 8) Select the following fields from all orders (trans\_date, price, promo\_code) renaming the colum (field) headings ('Transaction Date', 'Price' & 'Promotion Code')

-- 9) Select the following fields (customer\_surname, customer\_firstname, county) from all orders, with customer names in a single field named 'Customer Name' and in the format <Surname>, <Firstname>, with surname capitalised. The county field is to be renamed 'County'.

-- 10) Select the average price, minimum price & maximum price for each category.

-- 11) Select the category\_name (labelled 'Category', number of sales (labelled 'Total Orders') & total sales (labelled 'Total Sales') for each category.

-- 12) List all orders with the following fields (with the labls given) orders.trans\_date('Transaction Date'), categories.category\_name('Category'), orders.customer\_surname('Surname'), orders.customer\_firstname('Firstname'), orders.price('Order Price'), categories.category\_name('Category'), promotions.discount('Discounted by')

1. SELECT \* FROM `orders` where category\_id = 2

2. SELECT \* FROM `orders` WHERE `category\_id` IN (2,4,5)

3. SELECT \* FROM `orders` WHERE `price` > '35'

4. SELECT \* FROM `orders` WHERE `date\_of\_birth` < '1980-01-01' AND `newsletter` = '1'

5. SELECT \* FROM `orders` where customer\_surname='Davenport'

6. SELECT \* FROM `orders` where customer\_firstname like 'Br%'

7. SELECT \* FROM `orders` WHERE `category\_id` = 3 ORDER BY `price` DESC

8. SELECT `trans\_date` as 'Transaction Date', `price` as 'Price', `promo\_code` as 'Promotion Code' FROM `orders`

9. SELECT CONCAT(UPPER(`customer\_surname`), ' ', `customer\_firstname`) as 'Customer Name', `county` as 'Country' FROM `orders`

10. SELECT `category\_id`, AVG(`price`), MIN(`price`), MAX(`price`) FROM `orders` GROUP BY `category\_id`

11. SELECT `category\_name` as 'Category', COUNT(id) as 'Total Orders', SUM(price) as 'Total Sales' FROM orders GROUP BY `category\_id` INNER JOIN `categories` ON (orders.category\_id=categories.id);

12. SELECT `trans\_date` as 'Transaction Date', `price` as 'Order Price', `customer\_surname` as 'Surname', `customer\_firstname` as 'Firstname', `category\_name` as 'Category', `discount` as 'Discounted by' FROM (`orders` LEFT JOIN `categories` ON (orders.category\_id=categories.id)) LEFT JOIN `promotions` ON (orders.promo\_code=promotions.code)