DQL Tasks

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

SELECT count(\*) FROM `orders` WHERE `category\_id`=2; 23

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

SELECT count(\*) FROM `orders` WHERE `category\_id`IN(2,4,5);66

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

SELECT count(\*) FROM `orders` WHERE `price`>35; 71

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

SELECT count(\*) FROM `orders` WHERE `date\_of\_birth`<1980-01-01 AND `newsletter`=1; 35

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

SELECT\*FROM `orders` WHERE `customer\_firstname` OR `customer\_surname`="Davenport"; 7

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

SELECT\*FROM `orders` WHERE `customer\_firstname`like"Br%"order by `product\_id`; Brendan

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

SELECT\*FROM `orders` WHERE `category\_id`=3 order by `price`DESC;

| id | trans\_date | product\_id | category\_id | price | county | customer\_surname | customer\_firstname | date\_of\_birth | newsletter | promo\_code |

| --- | ---------- | ---------- | ----------- | ----- | -------------- | ---------------- | ------------------ | ------------- | ---------- | -------------- |

| 6 | 2019-09-02 | 90 | 3 | 99.12 | Ross-shire | Hooper | Xandra | 1975-01-22 | 0 | |

| 7 | 2019-09-19 | 89 | 3 | 98.06 | Midlothian | Leonard | Velma | 1970-12-05 | 0 | PROMO10 |

| 32 | 2019-08-14 | 37 | 3 | 95.56 | Bedfordshire | Warren | Nicole | 2004-03-28 | 1 | PROMO15 |

| 71 | 2019-08-17 | 22 | 3 | 88.35 | Berkshire | Bender | Declan | 1987-04-11 | 0 | |

| 37 | 2019-02-18 | 91 | 3 | 83.27 | Sutherland | Roth | Ebony | 2005-01-01 | 0 | |

| 16 | 2019-05-16 | 32 | 3 | 76.93 | Denbighshire | Cleveland | Amethyst | 1999-03-31 | 1 | |

| 67 | 2019-08-26 | 83 | 3 | 72.03 | Yorkshire | Velez | Martha | 1996-04-09 | 0 | |

| 9 | 2019-11-01 | 66 | 3 | 63.58 | Caithness | Velez | Griffin | 1938-08-05 | 0 | PROMO15 |

| 97 | 2019-08-05 | 52 | 3 | 62.49 | Yorkshire | Riggs | Veronica | 2004-02-02 | 0 | PROMO15 |

| 66 | 2019-09-07 | 6 | 3 | 6.75 | Kent | Barlow | Harriet | 1990-12-15 | 0 | |

| 26 | 2019-11-11 | 13 | 3 | 49.72 | Suffolk | Golden | Martin | 1974-05-26 | 0 | |

| 24 | 2019-07-11 | 78 | 3 | 45.14 | Norfolk | Schneider | Thor | 1942-05-28 | 1 | |

| 80 | 2019-08-18 | 53 | 3 | 39.49 | Suffolk | Davenport | Tyrone | 1989-03-16 | 1 | PROMO15 |

| 72 | 2019-04-01 | 38 | 3 | 38.74 | Yorkshire | Schmidt | Tatyana | 1936-06-23 | 0 | PROMO15 |

| 53 | 2019-06-15 | 1 | 3 | 28.61 | Angus | Carr | Joelle | 1999-04-09 | 1 | |

| 38 | 2019-08-12 | 92 | 3 | 27.15 | Midlothian | Adkins | Hilel | 1956-08-14 | 0 | |

| 65 | 2019-04-20 | 42 | 3 | 23.98 | Cumberland | Byers | Xerxes | 1950-08-26 | 0 | |

| 52 | 2019-11-11 | 70 | 3 | 20.77 | Cambridgeshire | Bruno | Jordan | 1997-06-05 | 0 | SUMMER MADNESS |

| 61 | 2019-08-16 | 27 | 3 | 2.84 | Cardiganshire | Reilly | Aileen | 1950-12-31 | 1 | PROMO15 |

| 12 | 2019-03-26 | 65 | 3 | 16.36 | Berwickshire | Carr | Grace | 1934-08-10 | 0 | |

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

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

-- 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'.

SELECT CONCAT(`customer\_firstname`, ' ', `customer\_surname`) as `Customer Name`,`county` as `County` FROM `orders`;

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

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

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

SELECT `category\_id`, SUM (product\_id) as 'Total Orders', SUM (product\_id \* price) as 'Total Sales' FROM `orders` GROUP BY `category\_id`

-- 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')

SELECT \* FROM (`orders` o JOIN `promotions` p ON ( o.promo\_code = p.code )) JOIN `categories` c ON ( o.category\_id = c.id );