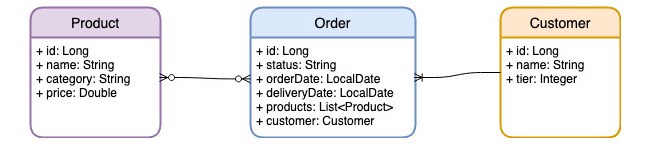
**Solve the below problem statements in Java: (use collections where ever it is necessary)**

Refer to the entity relationship diagram below, customers can place multiple orders and so it is a one-to-many relationship while the relationship between products and orders is many-to-many



Create respective classes as per above diagram.

Create a collection to store data\* related to Product,Order and Customer.

Create main program where logic for terminal and non terminal operations will be written as per below requirement:

* Obtain a list of products belongs to category “Books” with price > 100
* Obtain a list of order with products belong to category “Baby”
* Obtain a list of product with category = “Toys” and then apply 10% discount
* Obtain a list of products ordered by customer of tier 2 between 01-Feb-2021 and 01-Apr-2021
* Get the cheapest products of “Books” category

**\*Sample Data is given here**

**Customer(1, 'Stefan Walker', 1);**

**Customer(2, 'Daija Von', 1);**

**Customer(3, 'Ariane Rodriguez', 1);**

**Customer(4, 'Marques Nikolaus', 2);**

**Customer(5, 'Rachelle Greenfelder', 0);**

**Customer(6, 'Larissa White', 2);**

**Customer(7, 'Fae Heidenreich', 1);**

**Customer(8, 'Dino Will', 2);**

**Customer(9, 'Eloy Stroman', 1);**

**Customer(10, 'Brisa O''Connell', 1);**

**Product(1, 'omnis quod consequatur', 'Games', 184.83);**

**Product(2, 'vel libero suscipit', 'Toys', 12.66);**

**Product(3, 'non nemo iure', 'Grocery', 498.02);**

**Product(4, 'voluptatem voluptas aspernatur', 'Toys', 536.80);**

**Product(5, 'animi cum rem', 'Games', 458.20);**

**Product(6, 'dolorem porro debitis', 'Toys', 146.52);**

**Product(7, 'aspernatur rerum qui', 'Books', 656.42);**

**Product(8, 'deleniti earum et', 'Baby', 41.46);**

**Product(9, 'voluptas ut quidem', 'Books', 697.57);**

**Product(10, 'eos sed debitis', 'Baby', 366.90);**

**Product(11, 'laudantium sit nihil', 'Toys', 95.50);**

**Product(12, 'ut perferendis corporis', 'Grocery', 302.19);**

**Product(13, 'sint voluptatem ut', 'Toys', 295.37);**

**Product(14, 'quos sunt ipsam', 'Grocery', 534.64);**

**Product(15, 'qui illo error', 'Baby', 623.58);**

**Product(16, 'aut ex ducimus', 'Books', 551.39);**

**Product(17, 'accusamus repellendus minus', 'Books', 240.58);**

**Product(18, 'aut accusamus quia', 'Baby', 881.38);**

**Product(19, 'doloremque incidunt sed', 'Games', 988.49);**

**Product(20, 'libero omnis velit', 'Baby', 177.61);**

**Product(21, 'consectetur cupiditate sunt', 'Toys', 95.46);**

**Product(22, 'itaque ea qui', 'Baby', 677.78);**

**Product(23, 'non et nulla', 'Grocery', 70.49);**

**Product(24, 'veniam consequatur et', 'Books', 893.44);**

**Product(25, 'magnam adipisci voluptate', 'Grocery', 366.13);**

**Product(26, 'reiciendis consequuntur placeat', 'Toys', 359.27);**

**Product(27, 'dolores ipsum sit', 'Toys', 786.99);**

**Product(28, 'ut hic tempore', 'Toys', 316.09);**

**Product(29, 'quas quis deserunt', 'Toys', 772.78);**

**Product(30, 'excepturi nesciunt accusantium', 'Toys', 911.46);**

**order(1, '2021-02-28', '2021-03-08', 'NEW', 5);**

**order(2, '2021-02-28', '2021-03-05', 'NEW', 3);**

**order(3, '2021-04-10', '2021-04-18', 'DELIVERED', 5);**

**order(4, '2021-03-22', '2021-03-27', 'PENDING', 3);**

**order(5, '2021-03-04', '2021-03-12', 'NEW', 1);**

**order(6, '2021-03-30', '2021-04-07', 'DELIVERED', 9);**

**order(7, '2021-03-05', '2021-03-09', 'PENDING', 8);**

**order(8, '2021-03-27', '2021-04-05', 'NEW', 4);**

**order(9, '2021-04-14', '2021-04-18', 'NEW', 10);**

**order(10, '2021-03-10', '2021-03-19', 'NEW', 8);**

**order(11, '2021-04-01', '2021-04-04', 'DELIVERED', 1);**

**order(12, '2021-02-24', '2021-02-28', 'PENDING', 5);**

**order(13, '2021-03-15', '2021-03-21', 'NEW', 5);**

**order(14, '2021-03-30', '2021-04-07', 'PENDING', 4);**

**order(15, '2021-03-13', '2021-03-14', 'DELIVERED', 5);**

**order(16, '2021-03-13', '2021-03-21', 'NEW', 1);**

**order(17, '2021-03-31', '2021-03-31', 'DELIVERED', 6);**

**order(18, '2021-03-25', '2021-03-31', 'PENDING', 9);**

**order(19, '2021-02-28', '2021-03-09', 'DELIVERED', 9);**

**order(20, '2021-03-23', '2021-03-30', 'NEW', 5);**

**order(21, '2021-03-19', '2021-03-24', 'DELIVERED', 9);**

**order(22, '2021-02-27', '2021-03-01', 'NEW', 5);**

**order(23, '2021-04-19', '2021-04-24', 'PENDING', 4);**

**order(24, '2021-03-24', '2021-03-24', 'DELIVERED', 1);**

**order(25, '2021-03-03', '2021-03-10', 'NEW', 1);**

**order(26, '2021-03-17', '2021-03-26', 'NEW', 10);**

**order(27, '2021-03-20', '2021-03-25', 'NEW', 1);**

**order(28, '2021-04-09', '2021-04-16', 'DELIVERED', 2);**

**order(29, '2021-04-06', '2021-04-08', 'PENDING', 1);**

**order(30, '2021-04-19', '2021-04-20', 'DELIVERED', 1);**

**order(31, '2021-03-03', '2021-03-04', 'NEW', 3);**

**order(32, '2021-03-15', '2021-03-24', 'DELIVERED', 2);**

**order(33, '2021-04-18', '2021-04-24', 'PENDING', 1);**

**order(34, '2021-03-28', '2021-03-28', 'NEW', 6);**

**order(35, '2021-03-15', '2021-03-17', 'NEW', 1);**

**order(36, '2021-03-04', '2021-03-08', 'DELIVERED', 2);**

**order(37, '2021-03-18', '2021-03-25', 'NEW', 8);**

**order(38, '2021-04-11', '2021-04-20', 'NEW', 8);**

**order(39, '2021-04-12', '2021-04-17', 'NEW', 9);**

**order(40, '2021-03-12', '2021-03-12', 'PENDING', 3);**

**order(41, '2021-02-24', '2021-02-26', 'NEW', 5);**

**order(42, '2021-04-08', '2021-04-14', 'DELIVERED', 9);**

**order(43, '2021-03-03', '2021-03-11', 'NEW', 3);**

**order(44, '2021-03-12', '2021-03-14', 'DELIVERED', 4);**

**order(45, '2021-04-01', '2021-04-06', 'DELIVERED', 1);**

**order(46, '2021-03-16', '2021-03-22', 'NEW', 10);**

**order(47, '2021-04-07', '2021-04-12', 'PENDING', 2);**

**order(48, '2021-04-05', '2021-04-06', 'NEW', 2);**

**order(49, '2021-04-10', '2021-04-13', 'NEW', 7);**

**order(50, '2021-03-18', '2021-03-21', 'NEW', 9);**