NUID 2nd Last Digit ending in 2 or 7
Your NUID:

-- Question 1 (6 points)

The Wang Data Science Institute conducts Data Management research. Please design a database to support its recruiting operations based on the business requirements listed below. Normalize your design to the 3rd Normal Form.

1) Applicants need to be tracked.

2) An applicant's specialties must be tracked. Specialties are all coded. A specialty has a standard code.

3) There are many interviewers available for screening applicants.

4) An applicant's interview history must be tracked. To be more specific, management wants to know what interviewer(s) have met with an applicant.

5) If an applicant has passed the screening, an offer will be extended to the applicant. The offer must be tracked.

6) There are some standard offer packages. A successful applicant will be extended one of the packages.

Normalize the design to the 3rd normal form. Create the ERD in your design tool. Submit the ERD.

-- Question 2 (2 points)

Use the referencing technique to design MongoDB documents which contain the data below. Submit the documents in JSON format.

CustomerID	AccountNumber	SalesOrderID	OrderValue
30000	10-4020-000245	46645	114198
30000	10-4020-000245	51124	87230
30000	10-4020-000245	55275	72873

```
-- Question 3 (2 points)

/*
Write a query to retrieve all dates which had
products of more than 7 unique colors ordered in them.

For example, in a date if the following products were sold:
      Product 1 in red, quantity sold 6
      Product 2 in green, quantity sold 8
      Product 3 in yellow, quantity sold 25
      Product 4 in red, quantity sold 20

Then, the sold products had 3 unique colors.

Exclude the products which don't have a color specified.

Return the date and the number of the unique colors sold
in that date. Sort the returned data by the number of
unique colors in desc.
*/




-- Question 4 (2 points)

/* Write a query to retrieve customers who have made at least an order
   before but never purchased a product which was more expensive
   than $100. Use the unit price of SalesOrderDetail as the purchase
   price of an item. Include the customer id and customer's total number
   of orders in the report.

   Sort the returned data by the customer id. */
```

```
-- Question 5 (3 points)

/*
Write a query to retrieve a customer's most valuable order.
The most valuable order has the highest totaldue amount. The totaldue
amount is stored in the SalesOrderHeader table. If there is a tie,
the tie needs to be retrieved.

Return only the customers who had an order greater than $10000.

Return the customer's id, last name, first name, the order id and
totaldue value, and the total order quantity for the order.
Sort the returned data by the customer's id.
*/
```