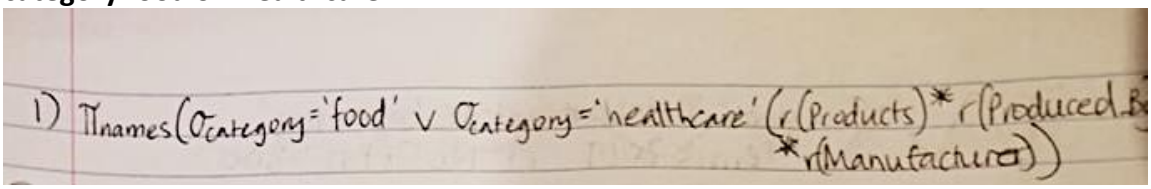
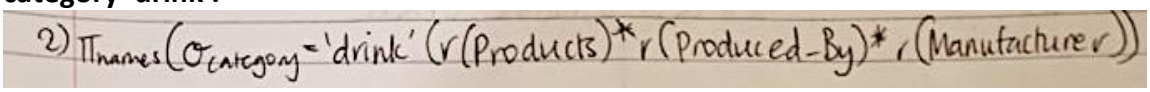
Question One - Relational Algebra

a) *Translate the following query into Relational Algebra:*

1) **Retrieve the names of all manufactures who produce some products of category food or 'healthcare'.**

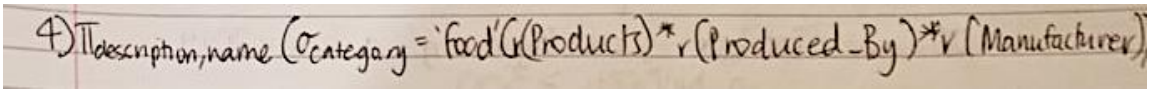


2) **Retrieve the names of all manufacturers who always produce products of category 'drink'.**



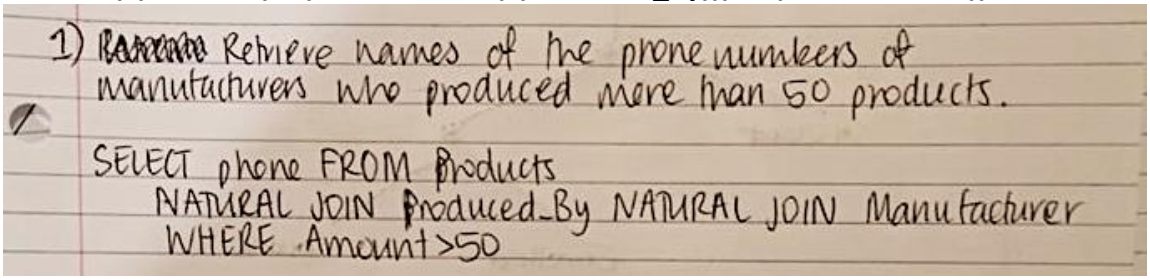
3) **Retrieve the descriptions of all products that are produced by two or more manufacturers.**

4) **For all products of category 'food' list their descriptions and the names of their manufacturers.**

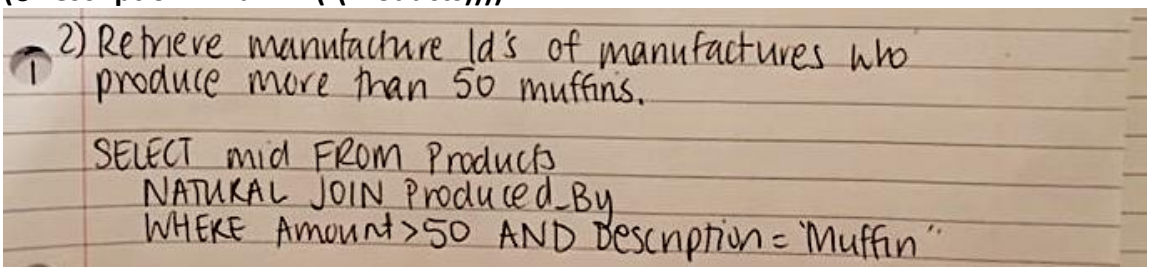


b) *Translate the following two queries into plain English and into SQL:*

1) **πPhone (r(Products) * (σAmount>50 (r(Produced_By))) * r(Manufacturer))**



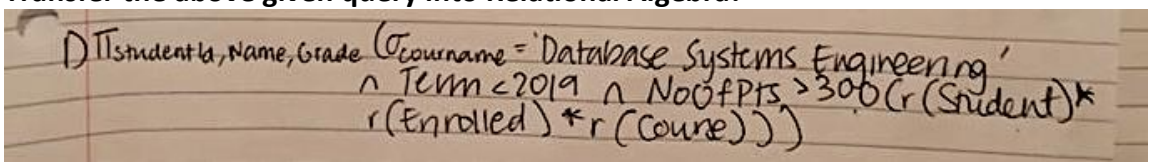
2) **πMId (σAmount>50 (r(Produced_By))) ∩ πMId (r(Produced_By) * (σDescription='Muffin' (r(Products))))**



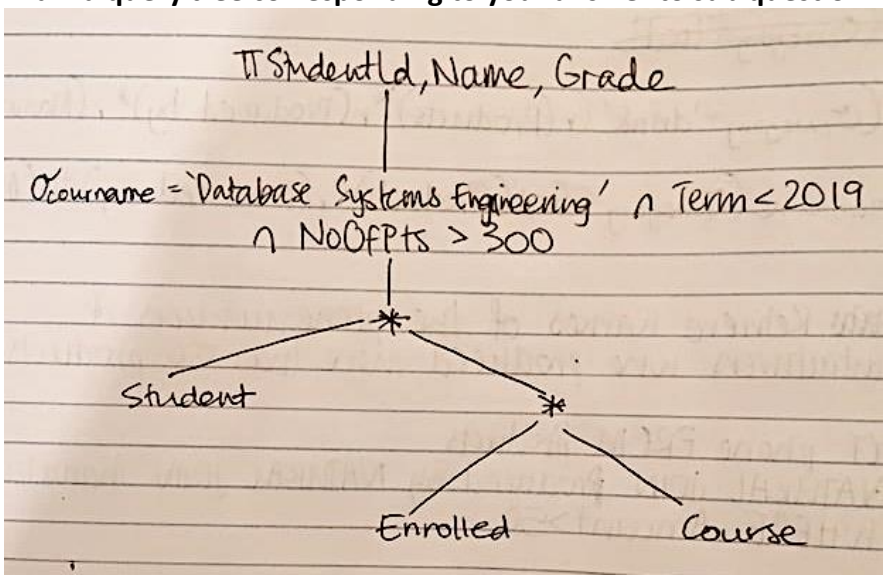
Question Two – Heuristic and Cost-Based Query Optimization

a) Heuristic query optimisation

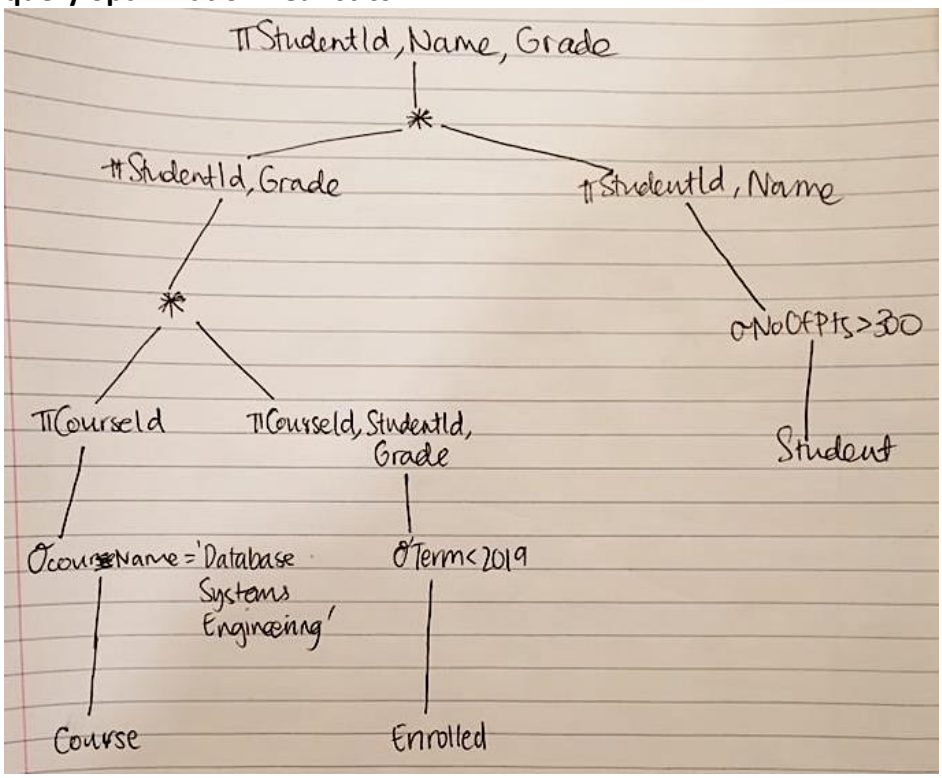
1) Transfer the above given query into Relational Algebra.



2) Draw a query tree corresponding to your answer to sub question 1)



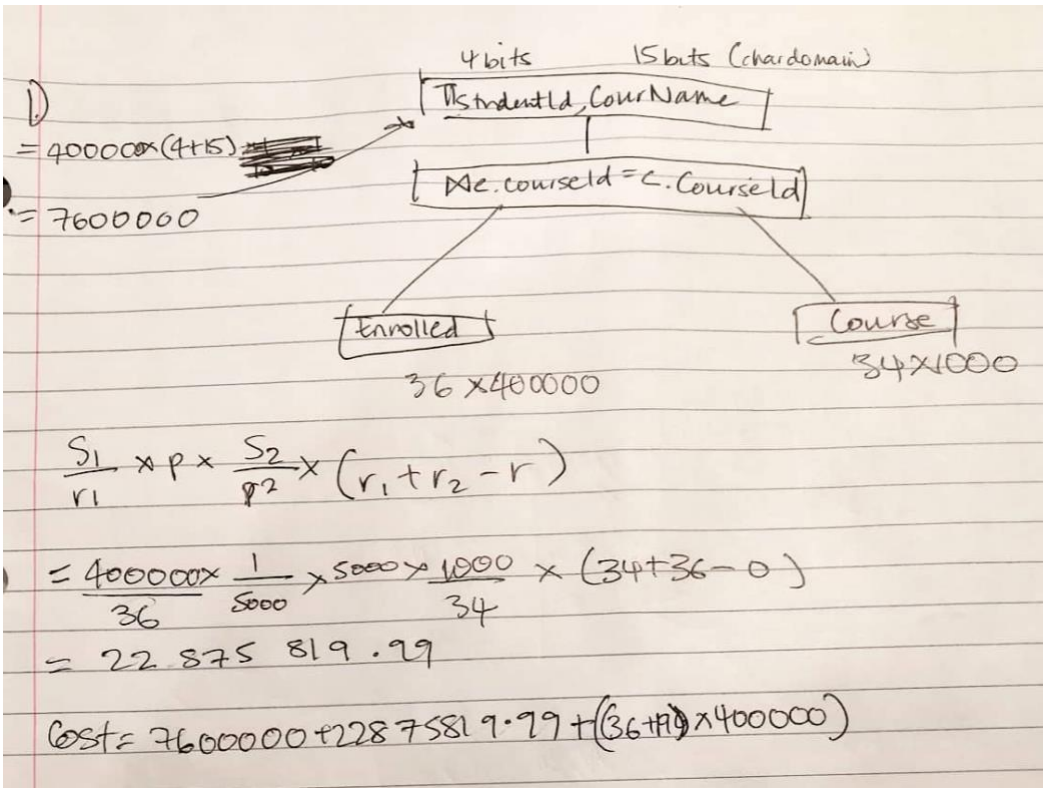
3) Transfer the query tree from 2) into an optimized query tree using the query optimization heuristics.



b) Query cost calculation

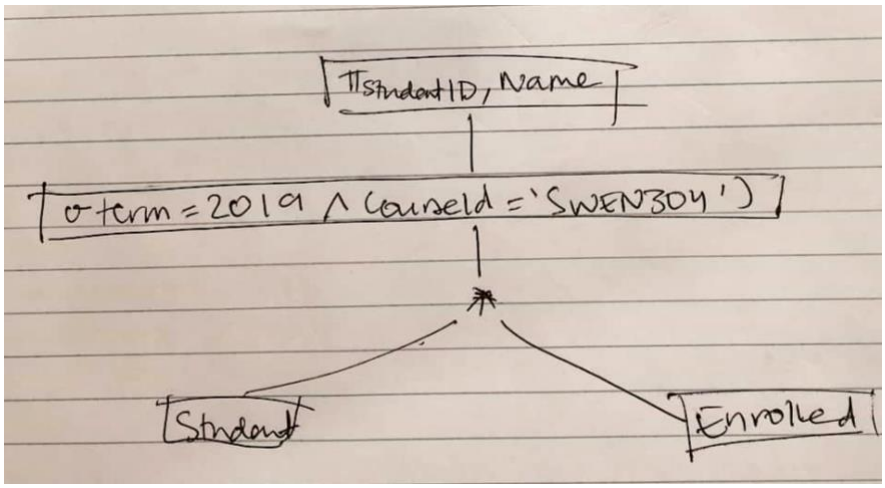
1) For the given query below, draw a query tree and calculate the cost of executing the query.

$$\pi_{\text{StudentId, CourName}} (r(\text{Enrolled}) \bowtie_{\text{e.CourseId = c.CourseId}} r(\text{Course}))$$



2) For the given query below, draw a query tree and calculate the cost of executing query.

$$\pi_{\text{StudentId, Name}} (\sigma_{\text{term =2019 } \wedge \text{ CourseId ='SWEN304'}} (r(\text{Student}) * r(\text{Enrolled})))$$



Question Three – PostgreSQL and Query Optimization

a) .

b) The original query gave this response:

```
 Seq Scan on customer  (cost=0.00..114.25 rows=1 width=56) (actual time=0.719..0.809 rows=1 loops=1)
   Filter: (customerid = 4567)
   Rows Removed by Filter: 4979
 Planning time: 0.485 ms
 Execution time: 0.908 ms
(5 rows)
```

I fixed this by using this query:

CREATE INDEX seach_customerID ON customer(customerID);
CREATE INDEX
EXPLAIN ANALYZE select l_name, f_name from customer where cutomerid = 4567;

```
  Index Scan using quick_search_customerid on customer  (cost=0.28..8.30 rows=1 width=32) (actual time=0.196..0.220 rows=1 loops=
1)
    Index Cond: (customerid = 4567)
  Planning time: 1.090 ms
  Execution time: 0.345 ms
 (4 rows)
```

By using this query it decreased my execution time as the index is smaller than an entire table and therefore will take less time to search through. (ref. https://solutioncenter.apexsql.com/how-to-create-and-optimize-sql-server-indexes-for-better-performance/).

c)