Exercise 1
Consider the TPC-H benchmark and the query:

```
select *
from lineitem l, orders o, customers c
where l.l_orderkey=o.o_orderkey
  and o.o_custkey=c.c_custkey
  and c.c_name="Customer#000014993".
```

Do canonical translation and logical optimization.

Exercise 2
Given \(|R_1|, |R_2|\), the domain of \(R_1.x\) and \(R_2.y\), and the information if \(R_1.x\) and/or \(R_2.y\) are keys of \(R_1\) and \(R_2\)

1. How can we estimate the selectivity of \(\sigma_{R_1.x=c}\), where \(c\) is a constant?

2. How can we estimate the selectivity of \(\Join_{R_1.x=R_2.y}\)?

Assume that the value distribution in both domains is uniform. Note that we don’t know the output size of \(\sigma_{R_1.x=c}\) (\(\Join_{R_1.x=R_2.y}\), respectively), so we can’t simply use the definition of selectivity.

Exercise 3
Given are two relations \(R\) and \(S\), with sizes 1,000 and 100,000 pages respectively. Each page has 50 tuples. The relations are stored on a disk, the average access time for the disk is 10 ms and the transfer speed is 10,000 pages/sec. How long does it take to perform the Nested Loops Join of \(R\) and \(S\)? How long does it take to perform the Block Nested Loops Join with a block size of 100 pages? Assume that CPU costs are negligible and ignore I/O costs for the join output.

Exercise 4
Using the program from the first exercise as a basis, implement a program that parses SQL queries, translates them into \(\text{tinydb}\) execution plans, and executes the query. Note: a canonical translation of the joins is fine, but push all predicates of the form \(attr = const\) down to the base relations.