Result size estimation:	**Cost estimation (# I/O) for single-relation plans**:
Result cardinality = Max # tuples * $\prod$ RF_i • col = value [example: $\sigma_{r.bid = 100}$] ➢ RF = $1/_{NDistinct(T1)}$ • col1 = col2 [example: $\sigma_{r.bid = s.bid}$] ➢ RF = $1/_{Max(NDistinct(T1), NDistinct(T2))}$ • col > value [example: $\sigma_{s.rating > 5}$] ➢ RF = $\frac{High(T1) - value}{High(T1) - Low(T1)}$	• Sequential scan of file: Cost = NPages(R) • B+ tree index I on key for equality search: Cost = Height(I) + 1 • Clustered index I for multiple select predicates: Cost = (NPages(I)+N**Pages**(R)) * $\prod RF_{matching}$ • Non-clustered index I matching one or more selects: Cost = (NPages(I)+N**Tuples**(R)) * $\prod RF_{matching}$

Tables:

Kitties: (kid [int], cuteness [1-10], owner [10 distinct]): 100 pages, 400 tuples

Puppies (pid [int], yappiness [1-10], owner [5 distinct]): 50 pages, 200 tuples

Humans: (hid [int], age [1-100]): 1,000 pages, 50,000 tuples

Indexes:

1. B+ tree (unclustered) on Kitties.cuteness [5 pages]
2. B+ tree (unclustered) on Puppies.yappiness [5 pages]
3. B+ tree (clustered) on (Puppies.owner, Puppies.yappiness) [15 pages]
4. B+ tree (unclustered) on Humans.hid [20 pages]

Query:

```
SELECT * FROM Kitties K, Puppies P, Humans H
WHERE K.owner = P.owner AND P.owner = H.hid
   AND P.yappiness = K.cuteness
   AND H.hid < 1200 AND P.yappiness = 7;
```

1. What are the best single-table plans (i.e., Phase 1)?

2. List the pairs of tables the optimizer will consider for 2-way joins (i.e., Phase 2)?

3. Which plans will be avoided?

Let's consider a possible sub-join of this expression, which in SQL would be

```
SELECT * FROM Kitties K, Puppies P
WHERE K.owner = P.owner
   AND P.yappiness = K.cuteness
   AND P.yappiness = 7;
```

4. What would be the IO cost of doing Index Nested Loops join using Puppies as the outer, with the optimal single table selection methods (see part 1)?

5. Now with Kitties as the outer.