Hash Index, Equality Selection
cost will be cost to find correct bucket.
+ # of I/Os to return correct tuples from that bucket (might be in several pages)

So 1 or 2 I/Os to find bucket

General Selects

Selection w/o Disjunction

Could ① do filescan to retrieve tuples
or use an index that matches some of conjuncts to scan records then cut from output according to non-primary conjuncts
② try to use several indexes

Idea sort rids satisfying conditions
represent using bit map
and maps

Selection w/ Disjunction

Consider

$$\sigma_{day < 8/9/02 \vee rname='Joe'}(R) = \varphi$$

no index   index

Because of this need to do file scan so no pt use index on rname.

However if have $\sigma_{\varphi \wedge sid=3}(R)$

index

Would use sid index to first winnow results down. The scan to calculate φ.

If had index on both day (B+) & rname then could use both indexes + bitmap strategy.

Projections Based on Sorting

Consider:

SELECT DISTINCT R.sid, R.bid
FROM Reserves R

Using Sorting:

(1) Scan R and produce a set of tuples that contain only the desired attributes

(2) sort this set

(3) Scan the sorted result, comparing adjacent tuples and eliminate duplicates

Cost: Scan Reserves 1000 I/Os If result tuples 10 bytes then tmp file write is 250 I/Os Sort step 1000 I/Os so 2500 I/Os

Scan sort set 250

Projections Based on Hashing

Suppose # of buffers pages B is large compared to # of pages in R.

Have one input buffer page B-1 out buffer

Read R into input project out unwanted attributes. Apply a hash f^1 to remainder to pick one of the B-1 output buffer

~~[illegible]~~

When an out buffer fills write to an out file

How does this help? Each out file partition contains different tuples. Within a partition tuples may or may not be different.

So we read in each partition now ~~[illegible]~~ and apply a second different hash f^2 ~~Slice~~. Provide $B > \sqrt{f \cdot T}$ can hash whole partition into existing pages in buffer. Then sort these pages & output.

$\left(\frac{T}{B-1}\right) \cdot f$ pages in partition tuples

What is cost

Read R

+ 2 Read tmp

= 1500 I/Os

Join Operations

Consider SELECT *
FROM Reserves R, Sailors S
WHERE R.sid = S.sid

simple join

Nested Loop Join

```
foreach page $p_R$ of R
    foreach page $p_S$ of S
        for all matching r, s in $p_R$ and $p_S$ add <r, s> to output
```

Cost if R is M pages
and S is N pages
then get $M + M \cdot N$

In running example $1000 + 1000 \cdot 500 = 501{,}000$ I/Os

At 10mS/I/O would take 1.4 hours.

- Note $R \bowtie B = B \bowtie R$

So what if used S in outer loop?

then get $N + M \cdot N = 500 + 1000 \cdot 500$
$= 500{,}500$ I/Os

So smaller relation should be in outer loop.