>>

COMP9315 Week 3 Exercises

- FYI
- Exercise: Building Tuples in PostgreSQL
- Exercise: How big is a **`FieldDesc`**?
- Exercise: Cost of Search in Hashed File
- Exercise: Update Operation Costs
- Exercise: Cost of n-Way Merge Sort
- Exercise: Cost of Relation Copy

^ >>

❖ FYI

Quiz 1 marks are now available

- normally available on Monday afer due date

Assignment 1

- make a new base data type (like **`integer`**)
- define: parser/printer, storage structure, operations
- new deadline 9pm Friday 19 March

Unix skills

- Home Computing playlist on

  https://www.youtube.com/channel/UCi3Kf5eONIwV6QgNHiYqVzg

<< ∧ >>

❖ Exercise: Building Tuples in PostgreSQL

Examine the code for

```
HeapTuple heap_form_tuple(desc,values[],isnull[])
```

and determine how a PostgreSQL tuple is built

```
HeapTuple = -> HeapTupleData
HeapTupleData =
   (t_len,t_self,t_tableOid,t_data->HeapTupleHeaderData)
HeapTupleHeaderData =
   (t_heap,t_ctid,t_infomask2,t_infomask,t_off,t_bits[],..)
HeapTupleFields =
   (t_xmin,t_xmax,(t_cid|t_xvac)
TupleDesc =
   (natts,tdtypeid,tdtypmod,tdhasoid,constraints[],atts[])
FormData_pg_attribute =
   (attrelid,attname,atttypid,attlen,attndims,attnotnull,...)
```

&lt;&lt; ∧ &gt;&gt;

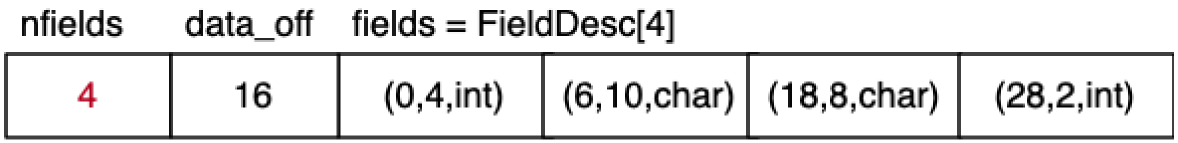
❖ Exercise: How big is a `FieldDesc`?

`FieldDesc` = (offset,length,type), where

- offset = offset of field within record data
- length = length (in bytes) of field
- type = data type of field

If pages are 8KB in size, how many bits are needed for each?

E.g.

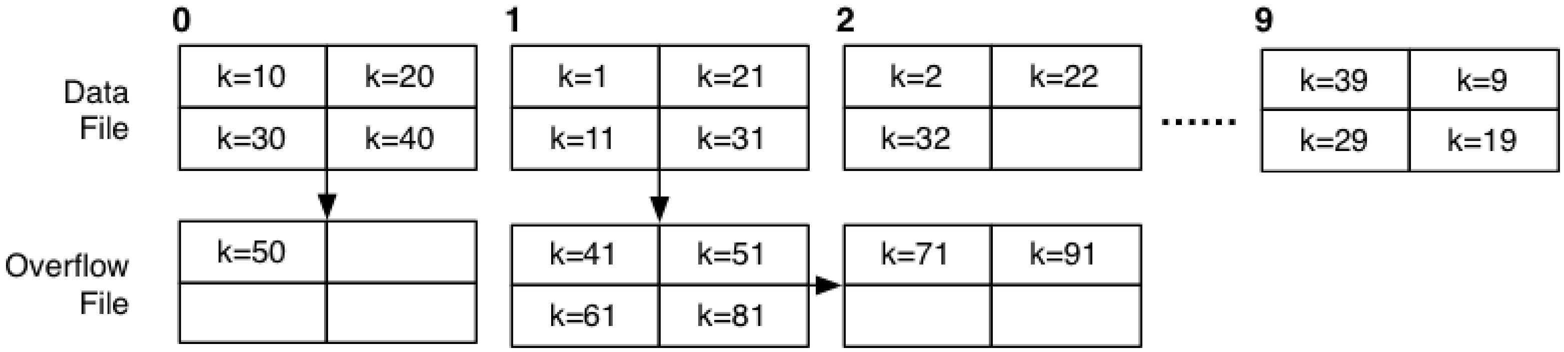


nfields	data_off	fields = FieldDesc[4]			
4	16	(0,4,int)	(6,10,char)	(18,8,char)	(28,2,int)

<< ∧ >>

❖ Exercise: Cost of Search in Hashed File

Consider the hashed file structure *b = 10, c = 4, h(k) = k%10*



Data File

0: k=10, k=20, k=30, k=40

1: k=1, k=21, k=11, k=31

2: k=2, k=22, k=32

......

9: k=39, k=9, k=29, k=19

Overflow File

k=50

k=41, k=51, k=61, k=81

k=71, k=91

Describe how the following queries

```
select * from R where k = 51;
select * from R where k > 50;
```

might be solved in a file structure like the above (*h(k) = k%b*).

Estimate the minimum and maximum cost (as #pages read)

<< ^ >>

❖ Exercise: Update Operation Costs

For each of the following file structures

- heap file, sorted file, hash file

Determine #page-reads + #page-writes for insert and delete

You can assume the existence of a file header containing

- values for *r*, *R*, *b*, *B*, *c*
- index of first page with free space (and a free list)

Assume also

- each page contains a header and directory as well as tuples
- no buffering (worst case scenario)

<< ^ >>

❖ Exercise: Cost of n-Way Merge Sort

How many reads+writes to sort the following:

- r = 1048576 tuples (2^{20})
- R = 62 bytes per tuple (fixed-size)
- B = 4096 bytes per page
- H = 96 bytes of header data per page
- D = 1 presence bit per tuple in page directory
- all pages are full

Consider for the cases:

- 9 total buffers, 8 input buffers, 1 output buffer
- 33 total buffers, 32 input buffers, 1 output buffer
- 257 total buffers, 256 input buffers, 1 output buffer

&lt;&lt; ∧

❖ Exercise: Cost of Relation Copy

Analyse cost for relation copying:

1. if both input and output are heap files
2. if input is sorted and output is heap file
3. if input is heap file and output is sorted

Assume ...

- *r* records in input file, *c* records/page
- b_{in} = number of pages in input file
- some pages in input file are *not* full
- all pages in output file are full (except the last)

Give cost in terms of #pages read + #pages written

Produced: 2 Mar 2021