Name ____________________________________________________________

Weight: 12.5% of final grade.
Total points possible: 110 (10 bonus points hidden throughout the exam)

**Budget your time carefully - you only get 1 hr 15 min for this exam.**

Unless otherwise requested, please explain the answers to the questions.

**Honor Pledge**

The University of Maryland Code of Academic Integrity requests that you write by hand and sign the following statement pledging your commitment to academic integrity. Please do so in the blank space below the text of the honor pledge.

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Signature ____________________________________________________________
A. Functional dependencies/normalization (40 pts)

1. Assume you are required to build a database storing information about a set of people. Specifically, for each person, you would keep track of:
   ● date of birth
   ● address
   ● gender
   ● occupation
   ● name

Assume you want to ensure this database satisfies the following requirements:
   1. Only one name is possible for a given date of birth (but multiple people may have the same name, i.e. the date of birth identifies the name, not the person).
   2. At a given address, the gender of a person uniquely identifies their date of birth. In other words, only one date of birth is possible for a given address and gender.
   3. The name and occupation of a person uniquely identify a person’s address.

Using the given schema and requirements, please answer the following questions:

A. (5 pts) Write down the set of functional dependencies described by the three requirements listed above.
B. (5 pts) What set of attributes can be used to uniquely identify a given person (tuple in your database)?
C. (10 pts) Assume you want to create a database whose tables are in the Boyce-Codd Normal Form. Indicate one possible decomposition of the original schema into BCNF. Please show your work: indicate which functional dependencies violate the BCNF condition and why, and describe the intermediate steps in determining the final decomposition.
D. (5 pts) Is the resulting schema (after decomposition) in the Third Normal Form? If yes, indicate why. If no, indicate a functional dependency that violates the 3NF conditions.
E. (5 pts) Is the original schema in 3NF? Explain why.
F. (10 pts) Assume that, in addition to the first 3 requirements, you also require that the address and gender uniquely identify a person’s occupation. Using this, together with the first three requirements, someone decided to decompose the original schema into:
   R1(date_of_birth, name), R2(date_of_birth, address, gender, occupation)

   Is this schema in BCNF? Explain why.

Hint: To make things easier, assign each attribute a letter, e.g. date of birth is A, address is B, etc.
Hint 2: The hardest part is computing the closure of the functional dependencies described above.
B. Storage/Files/Indexing (40 pts)

1. (10 pts) Consider the following sequence of page requests to a buffer manager that contains 3 pages: C,D, A,B,C,D,E, A,B
What are the current contents of the buffer pool assuming the buffer management policy is:
   a) MRU
   b) LRU

2. (10 pts) A file is being organized using static hashing over an integer search key, and the hash function used is $x \mod 3$ (so there will be 3 buckets), where $x$ is the search key. Show the result of inserting the following tuples starting with an empty index: 5, 6, 8, 101, 10. Assume each page can hold 2 records. For full credit draw the buckets and the values stored in them.

3. Consider the following B+ tree, where each leaf and internal node can have between 2 and 3 values.

```
1   6   8   |   12  14   |   22  28   |   31  37
12  22  31
```

a) (10 pts) Draw the tree resulting after the insertion of value 9.
b) (10 pts) Draw the tree resulting after the insertion of value 9 and removal of value 37.
C. Query processing (30 pts)

1. (10 pts) You are given a B+-Tree with order 100 that has 100000 total entries. Assume each leaf page stores exactly 10 records each (so the total number of pages occupied by the records = 100000/10 = 10000). You are to compute the total number of pages that need to accessed for a range query a < x < b, where x is the search key, and a and b are constants, that returns 1000 records.

2. (20 pts) Assume you are given the relation r(B, C) with the following statistics:
   number of tuples in r, \( n(r) = 10,000 \)
   number of blocks in r, \( b(r) = 100 \)
   number of tuples/block, \( f(r) = 100 \)
   selectivity (number of tuples retrieved for a value of attribute B), \( SC(B, r) = 1 \)
   number of distinct values of attribute B in r, \( V(B, r) = 10000 \)
   minimum value of attribute B, \( \min(B, r) = 0 \)
   maximum value of attribute B, \( \max(B, r) = 10000 \)

   a. (5 pts) Estimate the I/O cost for answering the query: \( \sigma_{B=100}(r) \), assuming the relation is not indexed.
   b. (5 pts) Assume relation r has a secondary B+-tree index on attribute B and that the entire tree fits in memory (i.e. you can find any individual tuple for free). What is the I/O cost for answering the query: \( \sigma_{B>9900}(r) \). Briefly explain.
   c. (5 pts) Assume relation r is indexed on attribute B by a primary hash index that fits entirely in memory. What is the I/O cost for answering the query: \( \sigma_{B>9000}(r) \). Briefly explain.
   d. (5 pts) Assume you had a choice between indexing the relation r on attribute B using either a B+-tree, or a simple sorted index. Which one would you choose if neither index could be entirely loaded into memory, and you knew you had to answer range queries? Please explain your choice.

Note: To insure yourself against arithmetic errors, please list the costs both as a function of \( n(r), b(r), \) etc., in addition to a single number representing the number of blocks accessed when executing the query. In addition, it helps if you indicate which part of the costs corresponds to retrieving the results and which part is associated with the cost of searching through the index.