Name: ________________

Account: CS186 - __ __ __

# CS186 Final Exam Spring 2016

*You should receive 1 double-sided answer sheet and a **21-page** exam. There are a total of **147 Points**. Mark your name and login on both sides of the answer sheet, and in the blanks above. For each question, place only your final answer on the answer sheet—do not show work or formulas there. You may use the blank spaces for scratch paper, but do not tear off any pages. You will turn in both question and answer sheets.*

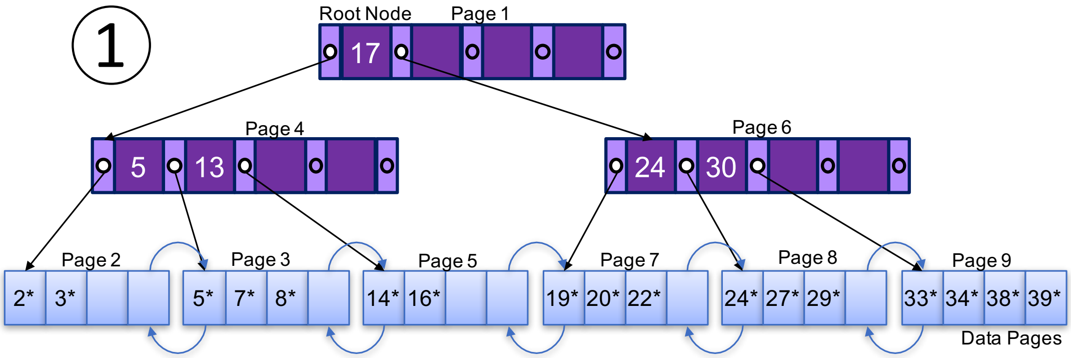
## I. Basics [14 Points]

For each image on the next page, select the letter corresponding to the best description.

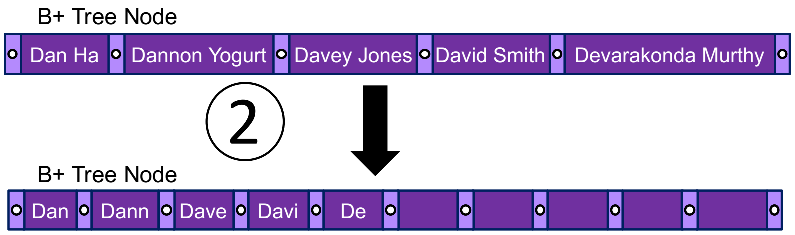
a) Left Deep Tree
b) Key Compression
c) B+Tree
d) ISAM
e) Nested Loops Join
f) Sort Merge Join
g) Indexed Nested Loop Join
h) Slotted Page
i) Variable Length Tuple
j) Fixed Length Tuple
k) Buffer Frame
l) Sort based group by
m) mapPartitions
n) Tournament Sort

This space is intentionally blank; exam continues on next page.

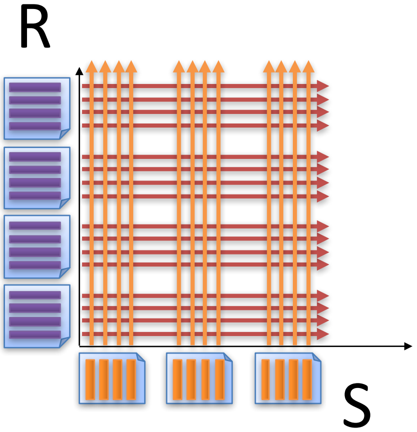
1



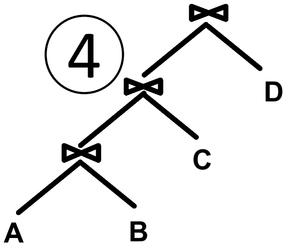
2



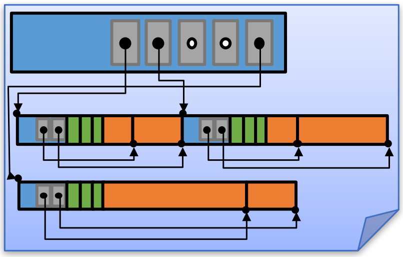
3



4



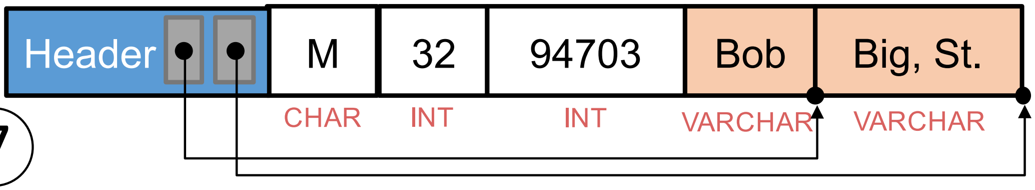
5



6

```
Sort(R)
Sort(S)
while not done {
  while (r < s) { advance r }
  while (r > s) { advance s }
  mark s
  while (R[r] == S[s]) {
    while (R[r] == S[s]) {
      yield <R[r], S[s]>
      advance s
    }
    reset s to mark
    advance r
  }
}
```

7



## II. Concurrency [13 Points]

1. **[3 Points]** For the following statements, mark the boxes for all true statement(s), or mark ∅ if none are true.
    a) In a system that uses two-phase locking, a transaction may perform writes after the first lock has been released.
    b) An SIX lock is compatible with another SIX lock on the same resource.
    c) In a system that uses the wound-wait protocol, there can be two disjoint deadlock cycles at the same time.
    d) By definition, a schedule is conflict serializable if it is equivalent to a serial schedule.
    e) In a system that uses strict two-phase locking, if a transaction aborts, it releases all of its locks as soon as rollback is complete.
    f) In a system that uses strict two-phase locking, a transaction that only performs reads can never enter a deadlock cycle.

Consider the following sequence of actions from time steps 1 through 10. The meanings of actions are shown below:

*R(A): the transaction reads tuple A* *W(A): the transaction writes tuple A*
*COM: the transaction commits*

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **T1** | R(A) | | | | W(A) | W(B) | | | COM | |
| **T2** | | R(A) | | | | | W(A) | | | COM |
| **T3** | | | W(B) | W(C) | | | | COM | | |

2. **[3 Points]** Draw the dependency graph for this schedule. (**Figure is on Answer Sheet.**)

3. **[2 Points]** Which of the following would guarantee that the schedule is serializable? Consider each scenario independently. Mark the boxes for all true statement(s), or mark ∅ if none are true.
    a) Remove T1's write to A at time step 5.
    b) Change the commit order (i.e., swap values in time-steps 8, 9, and 10)
    c) T1 and T2 write the same value to A at time steps 5 and 7, respectively.
    d) T1 writes a value to A that is one less than it read from A earlier; T2 writes a value to A that is 2 times larger than it read from A earlier.

Consider the following sequence of lock requests from time steps 1 through 6. Assume that the system uses **strict two-phase locking at only the tuple level**. None of the transactions commit or abort during the specified time period. The meanings of requests are shown below:

- *S(A): the transaction requests (but does not necessarily acquire) a shared lock on tuple A*
- *X(A): the transaction requests (but does not necessarily acquire) an exclusive lock on tuple A*

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|---|---|---|---|---|---|---|---|
| **T1** | S(A) | | | | | | |
| **T2** | | X(B) | | | | X(A) | |
| **T3** | | | | | S(A) | | |
| **T4** | | | X(C) | S(B) | | | |

4. **[3 Points]** Draw the waits-for graph for the series of requests above. (**Figure is on the answer Sheet.**)

5. **[1 Point]** (True/False) There are no deadlocks in the sequence of lock requests.

6. **[1 Point]** (True/False) If T1 requests a shared lock on B at time-step 7 there are **no deadlocks.**

This space is intentionally blank; exam continues on next page.

## III. Multigranularity Locking [6 points]

Consider the following sequence of actions from time steps 1 through 9. Assume that we are using strict two-phase locking with locks that can be acquired at the table, page, and tuple levels. Assume that at each time step, the transaction acquires the locks with minimal privilege required to complete the specified action.

Table Y contains pages A, B, and C. Page A contains tuples $A_1$, …, $A_{20}$. Page B contains tuples $B_1$, …, $B_{30}$. Page C contains tuples $C_1$, …, $C_{10}$. R(A) indicates that the transaction reads page A. W(A) indicates that the transaction writes page A. *Italicized and underlined* values indicate actions that were blocked because they could not acquire all of the necessary locks. Intention lock requests are not shown, but assume that the system follows multi-granularity strict two-phase locking. None of the transactions commit or abort during the specified time period.

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|
| **T1** | R($A_3$) | W(C) | | | | | *W($B_3$)* | | |
| **T2** | | | | | R(B) | R($A_1$) | | | |
| **T3** | | | | | | | | *R($C_1$)* | |
| **T4** | | | R($A_7$) | W($A_7$) | | | | | *W($B_3$)* |

1. **[2 points]** For each transaction write down the **table level locks** it has acquired on **table Y** at the moment T4 is blocked during time step 9. For instance, if a transaction acquires an S lock on Y, write S(Y); if none write ∅.

2. **[3 points]** Suppose T2 commits at time step 10. Which locks do T1, T3, and T4 **acquire** after this happens? Again, use locks with **minimal privilege**. For example, to indicate that a transaction acquires an S lock on page A, use the notation S(A). ***(****Hint: recall from Homework 4 that if multiple transactions are waiting to acquire a lock on a resource, we form a FIFO wait queue.)*

3. **[1 point] (True or False)** Consider time step 9 in the chart above. If T4's *W($B_3$)* at that time step had instead been R(A), then the action would **not** have been blocked.

## IV. ARIES RECOVERY [20 POINTS]

1. **[2 Points]** Suppose that you are forced to flush pages in the DPT to disk upon checkpointing. Which of the following cases are now guaranteed? **There is exactly one correct answer.**
   a) We can skip one of the three phases (analysis/redo/undo) completely
   b) We must start analysis from the beginning of the log
   c) Redo can start at the checkpoint.
   d) Redo must start from the beginning of the log
   e) Undo can start at the checkpoint
   f) Undo must run until the beginning of the log

2. **[4 Points]** Consider the following log, recovered after a crash. T1 and T2 are the only transactions. All pages were flushed to disk at and including LSN 60, so the log record has been truncated to start at LSN 70.

| LSN | Record | prevLSN |
|---|---|---|
| ... | .... | ... |
| 70 | UPDATE: T1 writes P3 | null |
| 80 | UPDATE: T2 writes P2 | 40 |
| 90 | Begin Checkpoint | - |
| 100 | End Checkpoint | - |
| 110 | UPDATE: T1 writes P1 | 70 |
| 120 | UPDATE: T1 writes P2 | 110 |
| 130 | T1 ABORT | 120 |
| 140 | CLR: T1 LSN 120, undoNextLSN: 110 | 130 |
| 150 | T2 COMMIT | 80 |
|  | ***CRASH*** |  |

   Assuming the buffer has **not** flushed any pages to disk after LSN 60, fill out the transaction table and dirty page table given on the answer sheet **as recorded in the end checkpoint record**. You may not need all rows, and may list the transactions/pages in any order.

3. **[6 points]** Now, fill out the transaction table and dirty page table on the answer sheet for the same log **after the analysis phase**. You may not need all rows.

4. **[6 points]** After recovery is complete, what records have been added to the log? You may not need all rows on the answer sheet. Be sure to follow the log record format we use above. (You may omit End records.)

5. **[1 point]** True/False: After recovery, both the Transaction Table and Dirty Page Table are empty.

6. **[1 Points]** Consider a slightly different scenario, in which the records at LSN 100 and 110 are in opposite order (that is, the UPDATE occurs before the End Checkpoint). Of the following, choose the **single** most appropriate answer.
   a) The DPT stored in the checkpoint must be different than in the original scenario.
   b) The DPT stored in the checkpoint could be different than in the original scenario.
   c) The DPT stored in the checkpoint must be the same as in the original scenario.

This space is intentionally blank; exam continues on next page.

## V. Two Phase Commit [14 points]

1. **[6 Points]** Mark the boxes for all true statement(s), or mark ∅ if none are true.
   a) Eventual consistency guarantees that if you read a copy from one replica, the other replicas will eventually be consistent with what you read.
   b) Two-phase commit (2PC) requires write-ahead logging, but not all log records must be flushed to disk before their corresponding message is sent.
   c) In 2PC, coordinators send prepare messages to participants. The coordinator must write a log record when all prepare messages have been sent.
   d) A global commit message cannot be sent before the coordinator has received a vote from all participants.
   e) A global abort message cannot be sent before the coordinator has received a vote from all participants.
   f) A participant machine that does not flush its prepare records to the log could violate the durability property.

2. **[2 Points]** Suppose that you are a machine that has just recovered from a crash. You discover a prepare record in your log for transaction *T*, and deduce that you are a participant. But you do not find a commit record. Under proper 2PC and logging protocols, mark the boxes for all true statement(s), or mark ∅ if none are true..
   a) The coordinator must have received your vote.
   b) You must determine if you need to commit or abort.

3. **[3 Points]** Given the scenario from the previous question, answer the following:
   a) On the answer sheet, circle all of the 2PC log records that **another participant** could have already logged.
   b) On the answer sheet, circle all of the 2PC log records that **the coordinator** could have already logged.

4. **[3 Points]** A participant machine can unilaterally (without consulting other machines) choose to abort at any time before it takes what action? On the answer sheet, choose exactly one of the sentences provided (mark the circle by your choice) and fill in the blank in your chosen sentence with the correct word. *Hint: Try to answer this question without looking at the options on the answer sheet, and then choose the option that fits your answer.*

This space is intentionally blank; exam continues on next page.

## VI. SQL & LINEAR ALGEBRA [11 PTS]

In lecture we learned that matrix algebra can be cast as operation on tables. Let's take a look linear algebra in SQL.

We have two *d*-dimensional sparse vectors (stored in row id, value form):

```
CREATE TABLE x (row INTEGER, value REAL)

CREATE TABLE y (row INTEGER, value REAL)
```

and two sparse matrices (stored in row id, col id, value form)

```
CREATE TABLE A (row INTEGER, col INTEGER, value REAL)

CREATE TABLE B (row INTEGER, col INTEGER, value REAL)
```

assume **A** is *n* by *d* and **B** is *d* by *m*.

1. **[2 Point]** On the answer sheet mark the letter corresponding to the SQL query that correctly computes the dot product

$$x \cdot y = \sum_{i=1}^{d} x_i y_i$$

of **x** and **y**. Assume they have compatible dimensions and **there is only one correct answer**.

a)
```
SELECT x.row AS row, x.value * y.value AS value
FROM x JOIN y
```
b)
```
SELECT SUM(x.value * y.value) AS value,
FROM x JOIN y ON x.row = y.row
```
c)
```
SELECT x.row, y.value, y.value
FROM x JOIN y ON x.row = y.row
```
d)
```
SELECT x.value * y.value
FROM x JOIN y ON x.row = y.row
```

2. **[2 Points]** Suppose we wanted to compute the elementwise sum of the vectors **x** and **y** using the SQL expression:

```
SELECT x.row AS row, sum(x.value + y.value) AS value
FROM x JOIN y
ON x.row = y.row
```

Which of the following statements about this query are **true**? Mark the boxes for all true statement(s), or mark $\varnothing$ if none are true.

a) Some non-zero entries may be omitted from the final result.
b) The correct query should use LEFT OUTER JOIN
c) The correct query should use FULL OUTER JOIN
d) There is nothing wrong

3. **[2 Points]** Which of the following expressions computes the matrix vector product:

$$(Ax)_i = \sum_{k=1}^{d} A_{ik} x_k$$

of the matrix **A** with the vector **x**? Assume **A** and **x** have compatible dimensions and **there is only one correct answer,**

a)
```
SELECT A.row AS row, A.value * x.value AS value
FROM A JOIN x
ON A.row = x.row
```
b)
```
SELECT A.row AS row, SUM(A.value * x.value) AS value
FROM A JOIN x
ON A.row = x.row
GROUP BY A.col
```
c)
```
SELECT A.row AS row, SUM(A.value * x.value) AS value
FROM A JOIN x
ON A.col = x.row
GROUP BY A.row
```
d)
```
SELECT x.row AS row, SUM(A.value * x.value) AS value
FROM A JOIN x
ON A.col = x.row
GROUP BY A.col
```

4. **[5 Points]** Fill in the blanks to complete the SQL implementation of matrix multiply between matrix **A** and **B** that is compute**:**

$$(AB)_{ij} = \sum_{k=1}^{d} A_{ik} B_{kj}$$

```
SELECT A.row AS row, B.col AS col,
       __________(a)__________ AS value
FROM A JOIN B
ON _____(b)_____ = _____(c)_____
GROUP BY _____(d)_____, _____(e)_____
```

Using the following options (you may any one option multiple times):

```
1) A.row
2) B.row
3) A.col
4) B.col
5) A.value + B.value
6) A.value * B.value
7) sum(A.value + B.value)
8) sum(A.value * B.value)
```

This space is intentionally blank; exam continues on next page.

## VII. Spark [15 pts]

1. **[6 Points]** Mark the boxes for all true statement(s), or mark ∅ if none are true..
   a) In the following block of code we know that **`a == 3 * b`**:

```
rdd = sc.parallelize(range(1, 1000))
a = rdd.flatMap(lambda x: [0, 1, 2]).count()
b = rdd.map(lambda x: [0, 1, 2]).count()
```

   b) `reduceByKey` and `reduce` both return an RDD.
   c) A call to `filter` may result in partitions with no records.
   d) The `mapPartitions` method takes a function from tuples to tuples
   e) `count()` forces the operations on the rdd to be evaluated.
   f) The following program will terminate

```
rdd = sc.parallelize(range(1, 1000))

def loop_forever(x):
      while true:
            print "I am thinking"
      return 42

profound_thought = rdd.map(loop_forever)

print "hello World"
```

This space is intentionally blank; exam continues on next page.

2. **[5 pts]** Given `rdd = sc.parallelize([1.0,2.0,3.0,4.0])`, compute the variance:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\mathbf{Var} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

by completing the following program:

```
xbar = rdd.____(a)____ / ____(b)____
var = rdd.____(c)_____.____(d)____ / ____(e)_____
```

For each letter in a blank, choose one of the following code snippets (you may use some snippets multiple times):

```
1. rdd.size()
2. rdd.count()
3. map(lambda x: x)
4. map(lambda x: (x - xbar) * (x - xbar))
5. mapPartitions(lambda x: (x - xbar) * (x - xbar))
6. reduce(lambda a, b: a + b)
7. reduceByKey(lambda a, b: a + b)
```

3. **[4 Point]** A friend at a big web company sent you the following snippet of Spark code from the core of the search engine:

```
data = [
   ("http://hello.com", "hello world ohai"),
   ("http://goodbye.com", "hello goodbye another day")
]
rdd = sc.parallelize(data)
rdd = rdd.flatMap(lambda x: [(y, [x[0]]) for y in x[1].split(' ')])
rdd.reduceByKey(lambda a, b: a + b).collect()
```

and asked you “What does it return”. **Choose one.**

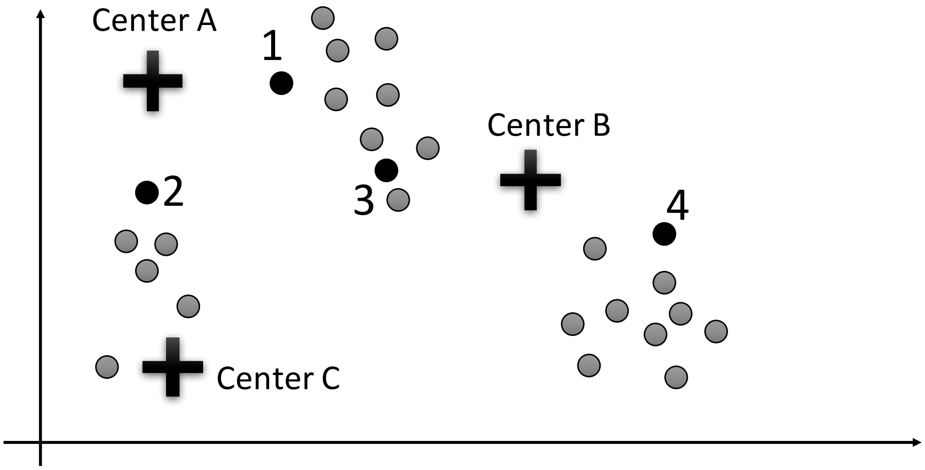
a. A page and the count of words in the page, e.g `("http://hello.com", 3)`
b. A page and the words with the frequency in that page, e.g. `("http://hello.com", [("hello", 1), …])`
c. A word and the frequency of that word across pages, e.g. `("hello", 2)`
d. A word and then the list of pages that contain that word, e.g. `("hello", ["http://hello.com", "http://goodbye.com"])`

## VIII. Analytics and Machine Learning [17 Points]

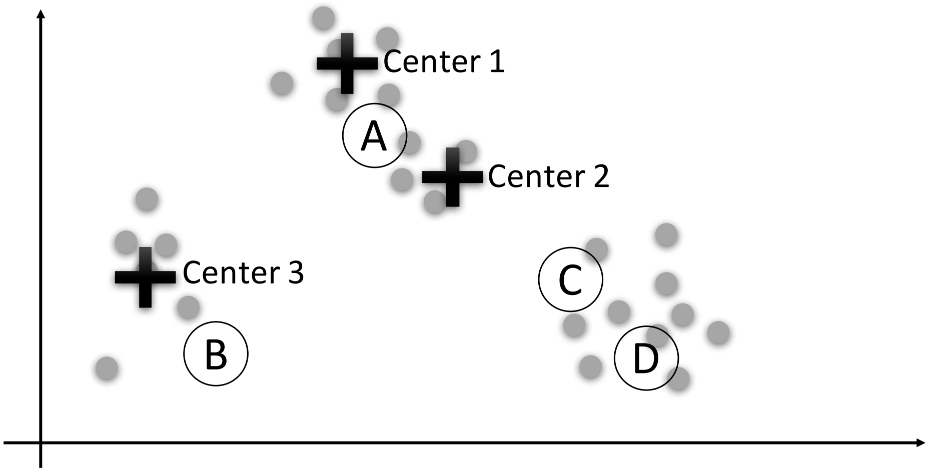
1) **[3 Points] Enterprise Data Infrastructure:** Mark the boxes for all true statement(s), or mark ∅ if none are true..
   a) A **schema-on-load** design implies that the schema is determined **when querying the data**.
   b) Data-lakes typically adopt a **schema-on-read** design to provide **greater flexibility** and simplify data collection.
   c) The **star schema** is a **normalized** representation of data in the data warehouse
   d) **Rollup** is the process of viewing data at increasingly **finer** granularity.
   e) In some situations storing the **messy or dirty raw data** can be **better than imposing a strict schema**.
   f) A multidimensional **data cube** can in fact have more than 3 dimensions.

2) **[3 Points] General Machine Learning**: Mark the boxes for all true statement(s), or mark ∅ if none are true.
   a) In **supervised** machine learning **labeled data** is used to train a model.
   b) **Classification models** would be a good candidate when trying to predict **the total amount of time** a user spends browsing a page.
   c) The ultimate goal in machine learning is **to find a model** that best fits the **training** data.
   d) The **k-means** algorithm is guaranteed **to converge** to the **optimal clustering**
   e) The **bag-of-words** model encodes **text as a vector**.
   f) A common form of **feature engineering** on **continuous** data is **one-hot-encoding**.

3) **[4 Points] K-Means Assignments:** For the following data choose the center assignments on the answer sheet for each of the points **1, 2, 3,** and **4**



4) **[3 Points] K-Means Cluster Centers:** For each of the cluster centers choose the letter on the answer sheet corresponding to the new location of that center after updating the assignments (one step of k-means). If the center doesn't move, choose DM (Doesn't Move).



5) **[2 Points] The A-Res Algorithm:** Using the following table containing the random numbers ***u*** and the calculated functions of ***u,*** mark the boxes of the ***2*** **letters** would be selected using the *A-Res* algorithm:

| **Letters** | **Weight (w)** | **u = rand()** | **u^(1/w)** | **u^w** | **w^u** |
|---|---|---|---|---|---|
| A | 1.0 | 0.641 | 0.641 | 0.641 | 1.000 |
| B | 2.0 | 0.504 | 0.710 | 0.254 | 1.418 |
| C | 4.2 | 0.396 | 0.802 | 0.020 | 1.765 |
| D | 1.3 | 0.253 | 0.347 | 0.168 | 1.069 |
| E | 5.9 | 0.973 | 0.995 | 0.851 | 5.624 |
| F | 0.7 | 0.212 | 0.109 | 0.338 | 0.927 |

6) **[2 Points] Regression and Overfitting:** Consider the following plots of training data, and functions fit to that data.

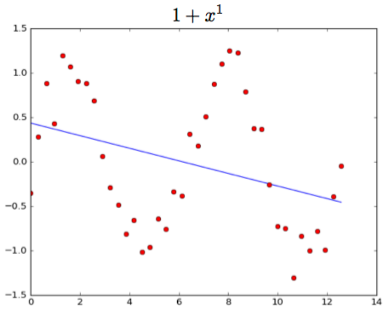


Figure A

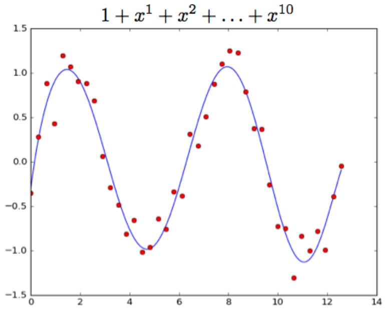


Figure B

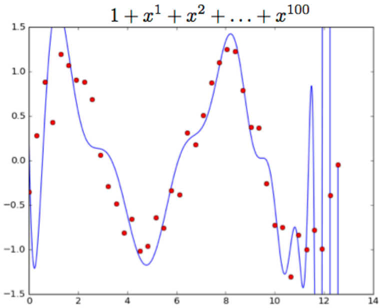


Figure C

a) Choose the letter corresponding to the plot of **over-fitting**

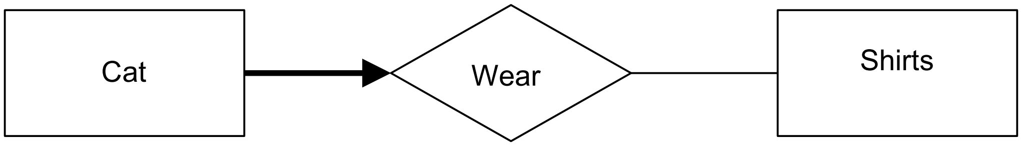
b) Choose the letter corresponding to the plot of **under-fitting**

This space is intentionally blank; exam continues on next page.

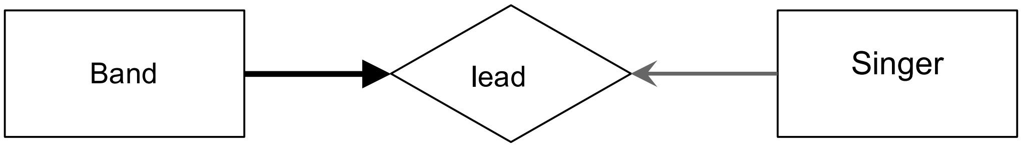
## IX. ER Modeling [14 points]

1. **[1 Points]** True/False: Any ternary relationship is equivalent to some pair of binary relationships.
2. **[3 Points]** For each of the following three diagrams, mark the circle by the **single** letter that corresponds to one of the following classifications of the relationship (read left-to-right):
   a) Many to Many
   b) One to Many
   c) Many to One
   d) One to One

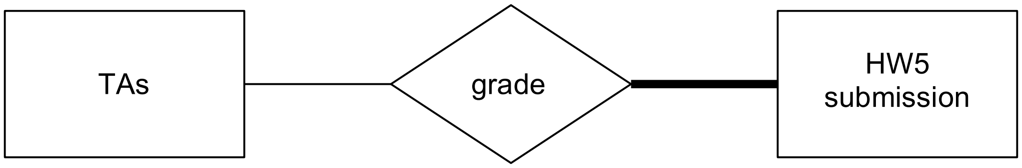
a.



b.



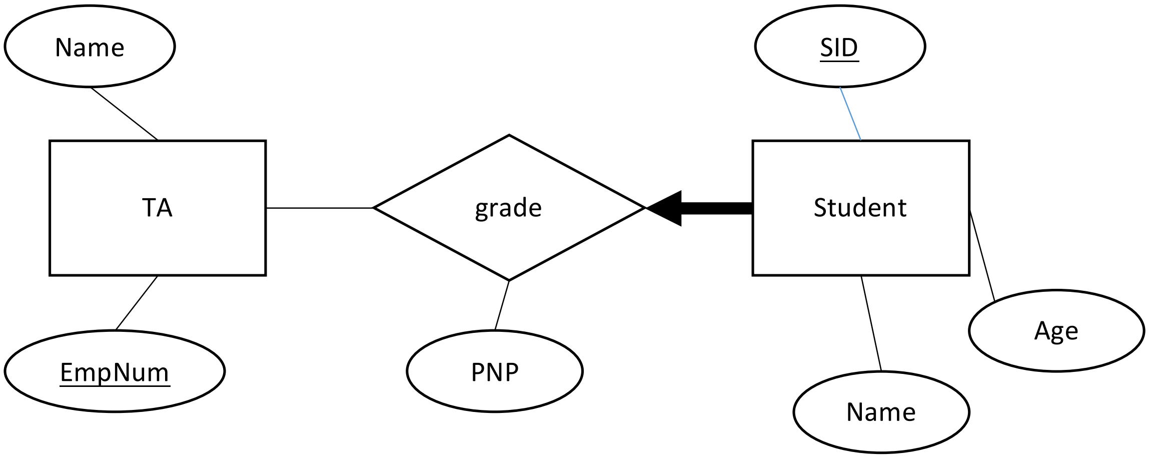
c.



This space is intentionally blank; exam continues on next page.

3. **[2 Points]** Given the following ER diagram, which SQL CREATE TABLE statements below best captures the Student entity set and grade relationship set correctly? **Choose exactly one.** For each choice, you can assume TA is defined as:

```
        CREATE TABLE TA(EmpNum integer PRIMARY KEY, Name text);
```



a.
```
CREATE TABLE Student(SID integer PRIMARY KEY, Age integer,
                     Name text);
CREATE TABLE grade(SID integer REFERENCES Student NOT NULL,
                   EmpNum integer REFERENCES TA,
                   PNP char(2),
                   PRIMARY KEY (SID, EmpNum));
```

b.
```
CREATE TABLE Student(SID integer PRIMARY KEY, Age integer,
                     Name text);
CREATE TABLE grade(SID integer REFERENCES Student NOT NULL,
                   EmpNum integer REFERENCES TA,
                   PNP char(2));
                   PRIMARY KEY (SID));
```

c.
```
CREATE TABLE StGrade(SID integer,
                     EmpNum integer REFERENCES TA NOT NULL,
                     Age integer, Name text, PNP char(2),
                     PRIMARY KEY (SID, EmpNum));
```

d.
```
CREATE TABLE StGrade(SID integer,
                     EmpNum integer REFERENCES TA NOT NULL,
                     Age integer, Name text, PNP char(2),
                     PRIMARY KEY (SID));
```

4. **[8 Points]** Please modify the edges on the ER diagram on the answer sheet to capture the following constraints. If you draw any bold lines, make sure they are clearly different from the non-bold lines.
   a) Each Donor can donate to multiple PACs but does not need to donate.
   b) Each PAC can receive donations from multiple donors and must receive at least 1 donation.
   c) Each Donor can be friends with multiple Politicians but does not need to be friends with any.
   d) Each Politician can be friends with multiple donors and needs to be friends with at least 1 donor.
   e) Each Donor can own multiple businesses but does not need to own any.
   f) Each corporation must be owned by exactly 1 donor.
   g) Each PAC can only run ads for 1 politician, but does not need to run any.
   h) Each politician can receive ads from many PACs but does not need to receive any.

This space is intentionally blank; exam continues on next page.

## X. Functional Dependencies and Normalization [15 points]

1. **[3 Points]** Check the boxes for all true statement(s), or check ∅ if none are true.
   a) Suppose A is a set of attributes over relation scheme R. Then it must be the case that: (A+)+ = A+.
   b) Any 2-column relation must be in BCNF.
   c) If A, B and C are attributes in table R with a functional dependency AB → C, then A → C and B → C.

2. **[4 Points]** Consider the relation R(A, B, C, D), with functional dependency set
   `F = {AB→C, BC→D, CD→A, AD→B, C→BD}`

   Check the boxes for all true statement(s), or check ∅ if none are true.
   a) C is a candidate key of R.
   b) D is a candidate key of R.
   c) AB is a candidate key of R.
   d) R is in BCNF.

3. **[4 Points]** Suppose you are given a relation R(A,B,C,D), with FDs F = {B → C, D → A}. Consider a decomposition of R into BC and AD. Check the boxes for all true statement(s), or check ∅ if none are true.
   a) The decomposition is lossy.
   b) One of BC and AD is not in BCNF.
   c) The decomposition is not dependency preserving.
   d) Neither BC nor AD has candidate keys.

4. **[4 Points]** Now consider R(A, B, C, D), decomposed into ACD and BC. Check the boxes for all true statement(s), or check ∅ if none are true.
   a) The decomposition is lossless if C is a candidate key for ACD
   b) The decomposition is lossless only if the rows of ACD and BC match up one-to-one.
   c) If the decomposition is lossy, the natural join of ACD and BC will be missing tuples found in R.
   d) If the decomposition is lossless, then both ACD and BC could have fewer rows than R.