Home Syllabus What's next Grades Reading list Help View your submissions View feedback / answers Logout

jmajikes

Assessment was due by Tue, 2022-12-06 19:10:00

Final Exam COMP 421 Fall 2022

Note: There are a total of 110 points on this exams.
You can only score a total of 100 points.
There is NO partial credit.

Don't panic!

You have 180 minutes to finish the exam.

- You *must* stay in full screen mode. Points removed for leaving full screen mode
- You must hand this final in on time.
    - Points removed for late submissions.
    - Only your **first** submission will be accepted.
    - Avoid accidental submissions. Fill in your name when you are ready to submit.
- Points removed for accidental submissions.

I recommend that you have several pieces of scrap paper to doodle notes on during the exam.

Consider this final **closed book**.
You MAY use your hand written notes. They MUST be on paper as you may not switch screens after starting the exam.

You **MAY NOT** Google or use other external websites for **answers** or copy from a friend. Do not paste information into your exam unless it was copied from your exam. You **MUST NOT** receive help from anyone.

If you do not know the origin of material you should not paste it into this exam. All material pasted into this exam must originate from this exam. This implies, but is not limited to, copying from previous assignments, copying from text messages, or copying from **any** website.

You **MUST** use the Google Chrome browser.

The browser will change input box color green to indicate correctness. A black or red box indicates an incorrect answer.

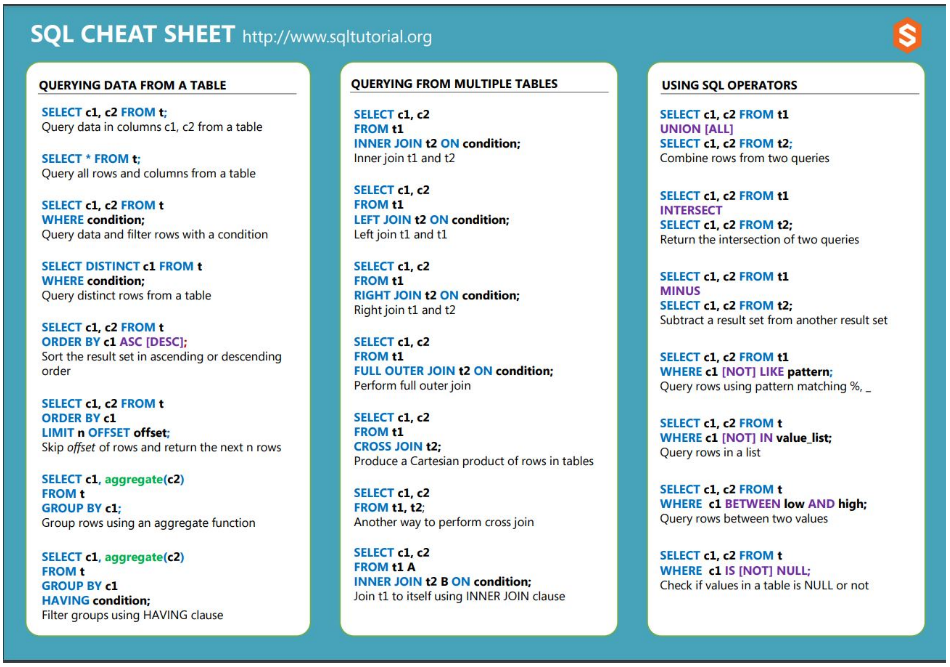
Note that HTML select statements with drop-downs are simple multiple choice questions. No highlighting of correct answers are done for select questions.

Green highlight should just assist you. If you believe your answer is correct and the input box did not turn green, continue on. Per the syllabus, highlighting is simply an aide not a guarantee.

Note: For database queries that are applied to two databases, **two** green lights are required to get any credit for the question.

SQL Tutorial Cheat Sheet

Following are three SQL tutorial cheat sheets available from http://www.sqltutorial.org



SQL CHEAT SHEET http://www.sqltutorial.org

QUERYING DATA FROM A TABLE

```
SELECT c1, c2 FROM t;
```
Query data in columns c1, c2 from a table

```
SELECT * FROM t;
```
Query all rows and columns from a table

```
SELECT c1, c2 FROM t
WHERE condition;
```
Query data and filter rows with a condition

```
SELECT DISTINCT c1 FROM t
WHERE condition;
```
Query distinct rows from a table

```
SELECT c1, c2 FROM t
ORDER BY c1 ASC [DESC];
```
Sort the result set in ascending or descending order

```
SELECT c1, c2 FROM t
ORDER BY c1
LIMIT n OFFSET offset;
```
Skip *offset* of rows and return the next n rows

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1;
```
Group rows using an aggregate function

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1
HAVING condition;
```
Filter groups using HAVING clause

QUERYING FROM MULTIPLE TABLES

```
SELECT c1, c2
FROM t1
INNER JOIN t2 ON condition;
```
Inner join t1 and t2

```
SELECT c1, c2
FROM t1
LEFT JOIN t2 ON condition;
```
Left join t1 and t1

```
SELECT c1, c2
FROM t1
RIGHT JOIN t2 ON condition;
```
Right join t1 and t2

```
SELECT c1, c2
FROM t1
FULL OUTER JOIN t2 ON condition;
```
Perform full outer join

```
SELECT c1, c2
FROM t1
CROSS JOIN t2;
```
Produce a Cartesian product of rows in tables

```
SELECT c1, c2
FROM t1, t2;
```
Another way to perform cross join

```
SELECT c1, c2
FROM t1 A
INNER JOIN t2 B ON condition;
```
Join t1 to itself using INNER JOIN clause

USING SQL OPERATORS

```
SELECT c1, c2 FROM t1
UNION [ALL]
SELECT c1, c2 FROM t2;
```
Combine rows from two queries

```
SELECT c1, c2 FROM t1
INTERSECT
SELECT c1, c2 FROM t2;
```
Return the intersection of two queries

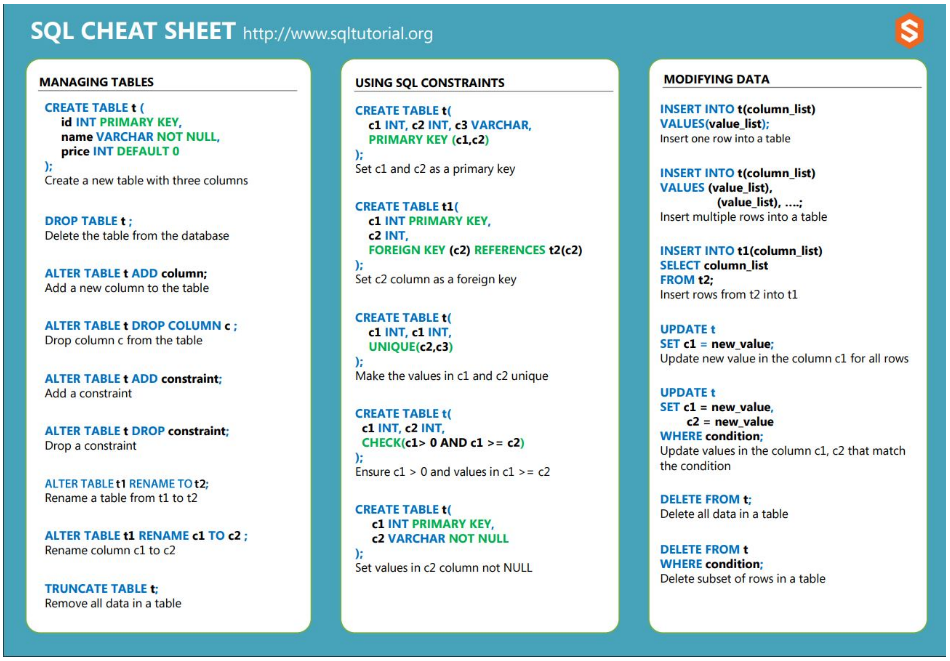
```
SELECT c1, c2 FROM t1
MINUS
SELECT c1, c2 FROM t2;
```
Subtract a result set from another result set

```
SELECT c1, c2 FROM t1
WHERE c1 [NOT] LIKE pattern;
```
Query rows using pattern matching %, _

```
SELECT c1, c2 FROM t
WHERE c1 [NOT] IN value_list;
```
Query rows in a list

```
SELECT c1, c2 FROM t
WHERE  c1 BETWEEN low AND high;
```
Query rows between two values

```
SELECT c1, c2 FROM t
WHERE  c1 IS [NOT] NULL;
```
Check if values in a table is NULL or not



SQL CHEAT SHEET http://www.sqltutorial.org

MANAGING TABLES

```
CREATE TABLE t (
    id INT PRIMARY KEY,
    name VARCHAR NOT NULL,
    price INT DEFAULT 0
);
```
Create a new table with three columns

```
DROP TABLE t ;
```
Delete the table from the database

```
ALTER TABLE t ADD column;
```
Add a new column to the table

```
ALTER TABLE t DROP COLUMN c ;
```
Drop column c from the table

```
ALTER TABLE t ADD constraint;
```
Add a constraint

```
ALTER TABLE t DROP constraint;
```
Drop a constraint

```
ALTER TABLE t1 RENAME TO t2;
```
Rename a table from t1 to t2

```
ALTER TABLE t1 RENAME c1 TO c2 ;
```
Rename column c1 to c2

```
TRUNCATE TABLE t;
```
Remove all data in a table

USING SQL CONSTRAINTS

```
CREATE TABLE t(
   c1 INT, c2 INT, c3 VARCHAR,
   PRIMARY KEY (c1,c2)
);
```
Set c1 and c2 as a primary key

```
CREATE TABLE t1(
   c1 INT PRIMARY KEY,
   c2 INT,
   FOREIGN KEY (c2) REFERENCES t2(c2)
);
```
Set c2 column as a foreign key

```
CREATE TABLE t(
   c1 INT, c1 INT,
   UNIQUE(c2,c3)
);
```
Make the values in c1 and c2 unique

```
CREATE TABLE t(
  c1 INT, c2 INT,
  CHECK(c1> 0 AND c1 >= c2)
);
```
Ensure c1 > 0 and values in c1 >= c2

```
CREATE TABLE t(
   c1 INT PRIMARY KEY,
   c2 VARCHAR NOT NULL
);
```
Set values in c2 column not NULL

MODIFYING DATA

```
INSERT INTO t(column_list)
VALUES(value_list);
```
Insert one row into a table

```
INSERT INTO t(column_list)
VALUES (value_list),
           (value_list), ….;
```
Insert multiple rows into a table

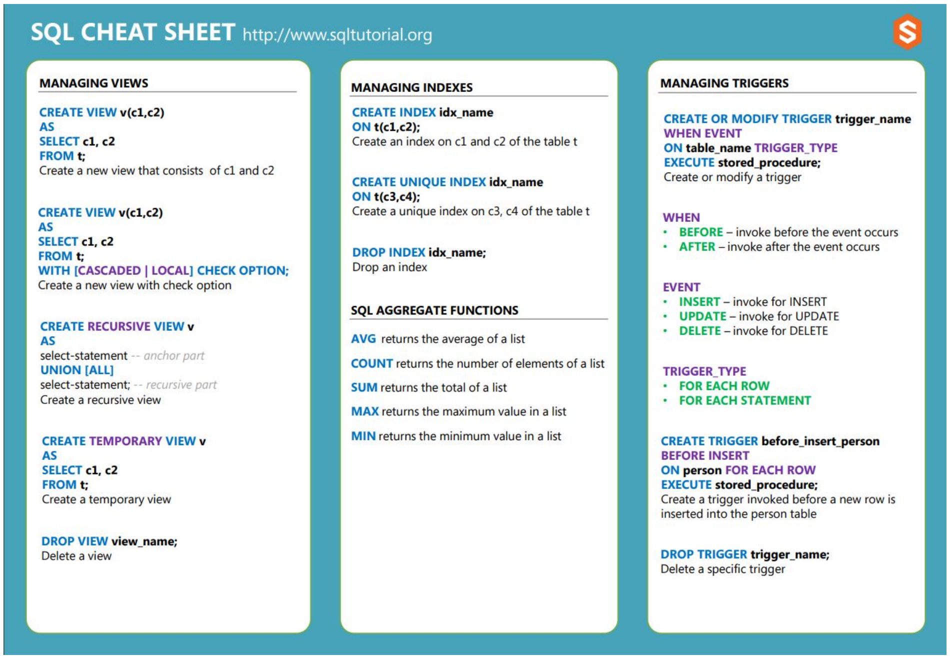
```
INSERT INTO t1(column_list)
SELECT column_list
FROM t2;
```
Insert rows from t2 into t1

```
UPDATE t
SET c1 = new_value;
```
Update new value in the column c1 for all rows

```
UPDATE t
SET c1 = new_value,
       c2 = new_value
WHERE condition;
```
Update values in the column c1, c2 that match the condition

```
DELETE FROM t;
```
Delete all data in a table

```
DELETE FROM t
WHERE condition;
```
Delete subset of rows in a table



SQL CHEAT SHEET http://www.sqltutorial.org

MANAGING VIEWS

```
CREATE VIEW v(c1,c2)
AS
SELECT c1, c2
FROM t;
```
Create a new view that consists of c1 and c2

```
CREATE VIEW v(c1,c2)
AS
SELECT c1, c2
FROM t;
WITH [CASCADED | LOCAL] CHECK OPTION;
```
Create a new view with check option

```
CREATE RECURSIVE VIEW v
AS
select-statement -- anchor part
UNION [ALL]
select-statement; -- recursive part
```
Create a recursive view

```
CREATE TEMPORARY VIEW v
AS
SELECT c1, c2
FROM t;
```
Create a temporary view

```
DROP VIEW view_name;
```
Delete a view

MANAGING INDEXES

```
CREATE INDEX idx_name
ON t(c1,c2);
```
Create an index on c1 and c2 of the table t

```
CREATE UNIQUE INDEX idx_name
ON t(c3,c4);
```
Create a unique index on c3, c4 of the table t

```
DROP INDEX idx_name;
```
Drop an index

SQL AGGREGATE FUNCTIONS

AVG returns the average of a list

COUNT returns the number of elements of a list

SUM returns the total of a list

MAX returns the maximum value in a list

MIN returns the minimum value in a list

MANAGING TRIGGERS

```
CREATE OR MODIFY TRIGGER trigger_name
WHEN EVENT
ON table_name TRIGGER_TYPE
EXECUTE stored_procedure;
```
Create or modify a trigger

WHEN
- BEFORE – invoke before the event occurs
- AFTER – invoke after the event occurs

EVENT
- INSERT – invoke for INSERT
- UPDATE – invoke for UPDATE
- DELETE – invoke for DELETE

TRIGGER_TYPE
- FOR EACH ROW
- FOR EACH STATEMENT

```
CREATE TRIGGER before_insert_person
BEFORE INSERT
ON person FOR EACH ROW
EXECUTE stored_procedure;
```
Create a trigger invoked before a new row is inserted into the person table

```
DROP TRIGGER trigger_name;
```
Delete a specific trigger

Sailor's Database Schema

Here are the tables you'll find for the Sailors database used in the final exam. Your queries will be run against two versions of the database. One of the databases will be much smaller and only contain a subset of the information.

```
CREATE TABLE Sailors (sid INTEGER PRIMARY KEY,
     sname TEXT,
     rating INTEGER,
     age INTEGER)
CREATE TABLE Boats (bid INTEGER PRIMARY KEY,
     bname TEXT,
     color TEXT)
CREATE TABLE Reserves (sid INTEGER,
     bid INTEGER,
     day TEXT,
     FOREIGN KEY (sid) REFERENCES Sailors(sid)
     ON DELETE CASCADE
     FOREIGN KEY (bid) REFERENCES Boats(bid)
     ON DELETE CASCADE)
```

Politician's Database Schema

Here are the tables you'll find for the politicians database used in the final exam. Your queries will be run against two versions of the database. One of the databases will be much smaller and only contain a subset of the information.

```
CREATE TABLE States
     (sid INTEGER PRIMARY KEY AUTOINCREMENT,
     state_name TEXT,
     state_abbreviation CHAR(2),
     state_capital TEXT,
     capital_latitude FLOAT,
     capital_longitude FLOAT,
     state_population INTEGER)
CREATE TABLE Politicians
     (pid INTEGER PRIMARY KEY AUTOINCREMENT,
     first_name CHAR(20),
     last_name CHAR(20),
     full_name TEXT,
     birthday DATE,
     gender CHAR(1))
CREATE TABLE Terms
     (tid INTEGER PRIMARY KEY AUTOINCREMENT,
     term_type TEXT,
     start_date DATE,
     end_date DATE,
     party CHAR(20),
     how_in_office CHAR(20),
     pid INTEGER,
     district INTEGER,
     sid INTEGER,
     FOREIGN KEY(sid) REFERENCES States(sid),
     FOREIGN KEY(pid) REFERENCES Politicians(pid))
```

Sailors scratch area

The following scratch space can be used to help develop and test queries against one of the **sailor** databases described above. The database used by the exam grader will be different.

Execute

Politicians scratch area

The following scratch space can be used to help develop and test queries against one of the **politicians** databases described above. The database used by the exam grader will be different.

```
SELECT DISTINCT term_type from terms
```

Execute

Questions For a total of 110 points

SQL Queries 65 out of 110 points

In this section, you will write SQL queries. Your queries will be tested immediately against two different databases. If your queries output matches the expected output, the displayed answers will be outlined in green. You need green, highlighted output from the query of both databases to get any credit. Your actual score will be determined when your query is tested against a different database but **two** green feedbacks should mean that you are on track to receive full credit.

Sailor.name.1: List the names of all the sailors.

Execute 1 points Minimize Output

Offices.1: List all the unique offices (senator, president, etc) held by all the politicians in the database.

Execute 1 points Minimize Output

Presidents.1: List the full name of all the U.S. Presidents in the order that they took office.
NOTE: Only list each president once. For example, Grover Cleveland was president before and after President Harrison. But Grover should only appear once.

Execute 1 points Minimize Output

Boat.Color.1: List all the boat colors.
NOTE: No boat color should appear more than once

Execute 1 points Minimize Output

Frodo.1: List all the boat names reserved by sailor Frodo.

Execute 1 points Minimize Output

All.Boats.1: List the names of all the sailors who reserved all the boats.

Execute 10 points Minimize Output

State.Names.1: List the following for the state names that alphabetically come before 'Missouri':

1. The state name
3. The state population

NOTE: List the states in alphabetically by state abbreviation.

Execute 10 points Minimize Output

State.Population.1: List the following for the senator(s) who started their term after January 1st, 2016 who are from the state(s) with the largest population:

1. Their first name
2. Their last name
3. Their political party

Execute 10 points Minimize Output

President.And.Senate.1: List two things about the Presidents of the United States:

1. Their full name
2. If they were a senator, the state name(s) they were a senator for. If they weren't a senator, `Null`.

Note: The approximately 45 presidents should be listed only once and by the date they first became president in chronological order.

Execute 10 points Minimize Output

Multi.Term.President.1: For all the presidents who served more than one term list:

1. Their full name
2. Their party affiliation

NOTE: List the information alphabetically by last name

Execute 10 points Minimize Output

State.Names.Capitals.1: List a single field relation that contains the set of state names and capitals.
Note: This should produce only one column. Both Raleigh and North Carolina should appear in different rows but both in the first column.

Execute 10 points Minimize Output

I/O Evaluations 35 out of 110 points

Counting Relations 5 out of the 35 i/o evaluations points out of 110 total points

Calculator: You may use this box as a calculator. Just type in any expression that can be evaluated by a JavaScript eval. For example, `2 ** 0.5` will show you the result of the square root of 2 in the red box to the right. Leaving the box empty or filled in will **NOT** affect your grade.

Enter an expression undefined

Use the following schema and table information for the next five one-point questions:

```
Sailors (sid: integer, sname: string, rating: integer, age: real)
Reserves (sid: integer, bid: integer, day: dates)
```

The Sailors table has 40,000 rows of tuples. Each row is 50 bytes, 80 rows fit on a page.
The Reserves table has 100,000 rows of tuples. Each row is 40 bytes, 100 rows fit on a page.

Cardinality.1: What is the cardinality of Reserves?

1 point

Btree.1: If Sailors has a B+tree index on ⟨ sname, rating ⟩, can it be used to help reduce the number of I/Os on the query `SELECT * FROM Sailors WHERE sname > 'J'`?

1 point

Hash.1: If Reserves has a hash index on sid can it be used to help reduce the number of I/Os on the query `SELECT * FROM Reserves WHERE sid < 3`?

1 point

Pages.1: Assuming 100% occupancy, how many pages does Reserves require?

1 point

Arity.1: What is the degree of Sailors?

1 point

Two way sort and external sort 10 out of the 35 i/o evaluations points out of 110 total points

Calculator: You may use this box as a calculator. Just type in any expression that can be evaluated by a JavaScript eval. For example, `2 ** 0.5` will show you the result of the square root of 2 in the red box to the right. Leaving the box empty or filled in will **NOT** affect your grade.

Enter an expression undefined

Note: Assume a 17 page file needs to be sorted and that 8 buffers are available for use.

Sort.Two.Way.Merge.Sort.1: Complete the following table about sorting the 17 page file using two way merge sort and at most 8 buffers.
If the pass doesn't get run select `N/A`.

Question	Answer
How many **buffers** are used for pass-0 of **two way merge sort**?	
How many **I/Os** are used for pass-0 of **two way merge sort**?	
What is the largest **run length** after pass-0 of **two way merge sort**?	
How many **runs** exist after pass-0 of **two way merge sort**?	
How many **passes** are required to sort using **two way merge sort**?	

5 points

Sort.External.Merge.Sort.1: Complete the following table about sorting the 17 page file using external merge sort and at most 8 buffers.
If the pass doesn't get run select `N/A`.

Question	Answer
How many **buffers** are used for pass-0 of an **external merge sort**?	
How many **I/Os** are used for pass-0 of an **external merge sort**?	
What is the largest **run length** after pass-0 of an **external merge sort**?	
How many **runs** exist after pass-0 of an **external merge sort**?	
How many **passes** are required to sort using **external merge sort**?	

5 points

Block Nested Loop Joins 10 out of 35 i/o evaluations points out of 110 total points

Calculator: You may use this box as a calculator. Just type in any expression that can be evaluated by a JavaScript eval. For example, `2 ** 0.5` will show you the result of the square root of 2 in the red box to the right. Leaving the box empty or filled in will **NOT** affect your grade.

Enter an expression undefined

Assume that Sailors is a file of 500 pages, Reserves is a file of 1000 pages, and there are 502 buffers.

Question	Answer
How many buffers are used for the outer loop?	
How many buffers are used for the input and output of the inner loop?	
Assuming Sailors is the outer loop, how many times is each Sailors page read in?	
Assuming Reserves is the outer loop, how many times is each Reserves page read in?	

Assuming Sailors is the inner loop, how many times is each Sailors page read in?	
Assuming Reserves is the inner loop, how many times is each Reserves page read in?	
Depending on which relation is the outer or inner loop, what is the optimal number of I/Os required to complete the BNL join	

10 points

Index Nested Loop Joins 10 out of 35 i/o evaluations points out of 110 total points

In this question and since selection has all integers, this question has a hash that requires 2 I/Os.

Assume the schema is:

```
CREATE TABLE Sailors (sid INTEGER PRIMARY KEY,
     sname TEXT,
     rating INTEGER,
     age INTEGER)
CREATE TABLE Boats (bid INTEGER PRIMARY KEY,
     bname TEXT,
     color TEXT)
CREATE TABLE Reserves (sid INTEGER,
     bid INTEGER,
     day TEXT,
     FOREIGN KEY (sid) REFERENCES Sailors(sid)
     ON DELETE CASCADE
     FOREIGN KEY (bid) REFERENCES Boats(bid)
     ON DELETE CASCADE)
```

Further assume you have to join the following:

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

Calculator: You may use this box as a calculator. Just type in any expression that can be evaluated by a JavaScript eval. For example, `2 ** 0.5` will show you the result of the square root of 2 in the red box to the right. Leaving the box empty or filled in will **NOT** affect your grade.

Enter an expression undefined

Assume that Sailors is a file of 20 pages, each page contains 80 tuples, Sailors has a hash index on `sid` that requires 2 I/Os to retrieve the data, and the data entries are stored using Alternative (2). You may assume that each sailor has reserved less than 100 boats.

Reserves is a file of 50 pages, each page contains 100 tuples, Reserves has a B+ tree index that requires 4 I/Os to retrieve the clustered data. To facilitate the math:
- The B+ tree is not buffered
- Each access of a data leaf through the index requires 4 I/Os.
- No sailor has reserved more than 50 boats.

Index.Nested.Loop.Join.Sailor.Outer.1: Answer the following questions about an index nested loop join with **Sailors** as the outside loop.

Question	Answer
How many I/Os to scan Sailors?	
How many Sailors tuples will be tested to match for a Reserve?	

How many I/Os to retrieve all the reserves by a single sailor?	
How many I/Os to complete the index nested loop join?	

5 points

Index.Nested.Loop.Join.Reserves.Outer.1: Answer the following questions about an index nested loop join with **Reserves** as the outside loop.

Question	Answer
How many I/Os to scan Reserves?	
How many Reserves tuples will be tested to match for a Sailors?	
How many I/Os to retrieve a single reserve's sailor?	
How many I/Os to complete the index nested loop join?	

5 points

B+ Trees (10 out of 110 points)

Figure 1 is a B+ Tree with index pages that can contain at most two keys and leaf pages that can contain at most two data entries. The root has two keys, 6 and 9, and three pointers.

After inserting the data with key 9 using the DBMS text book insertion algorithm from chapter 10, the resulting B+ tree will have the shape of Figure 2. Use the table at the right to fill in the key and data values for the B+ tree after the insertion. You must follow the DBMS book Chapter 10 algorithm for insertion into a B+ tree. (This means, do not do redistribution of nodes which was not covered in class.)

For example, if you believe that after inserting 9 into the B+ tree in Figure 1, then the resulting root′ would be have a key value of 9, then in the table for **Variable name in tree** row **A**, put the value 9. For any values that are empty, enter `null`.

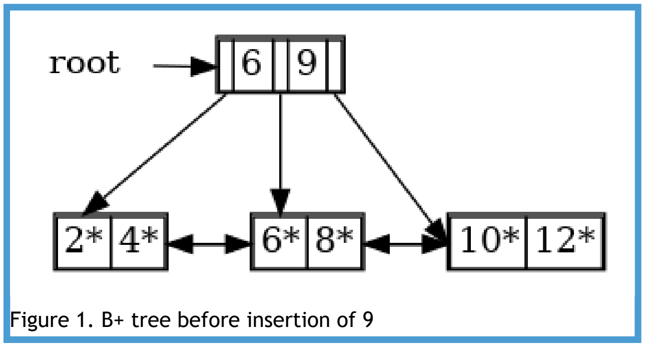


Figure 1. B+ tree before insertion of 9

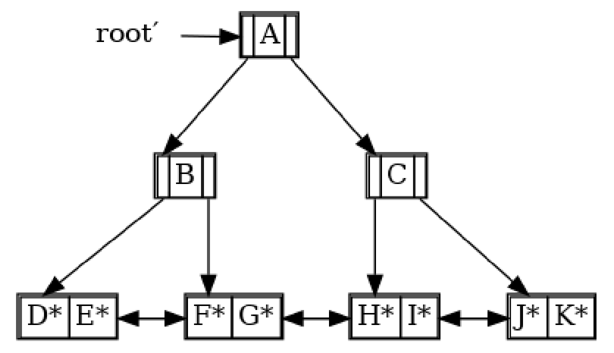


Figure 2. B+ tree after insertion of 9 with variables for key and data values

Btree.After.Insertion.1: Fill in the key and data values for the resulting B+ tree.
Note: Even though, alternate 2 leaf pages include an asterisk in the notation, the pull downs do not include the asterisk.

Variable name in tree	Value
A	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	

10 points

Pledge your exam

DO NOT PUT YOUR NAME ON THE EXAM UNTIL YOU ARE READY TO SUBMIT IT.
If you submit it and decide you would like to invalidate a submission, it will cost 10 percent of the points!

UNC Honor Pledge: I certify that no assistance has been received or given in the completion of this work. This unauthorized assistance includes, but is not limited to, using Google or other external websites, copying another student's answers, or giving another student answers.

john

If other students helped in the completion of this exam, and you would like to avoid an Honor Court hearing, give attribution to them by entering their Onyen below. Of course, getting/giving assistance will result in your submission being invalidated but the attribution will avoid an Honor Court hearing.
If no other student helped with the completion of this exam, **None** must be entered below.

majikes

Done !

Submit your work

Only your **first** submission will be accepted.

Did you **pledge your work** above?

Did you **acknowledge any student assistance** above?

Submit