Assessment was due by Thu, 2023-04-06 00:00:00

Midterm 2 COMP 421 Spring 2023

Note: There are a total of 110 on this exam. The highest grade you can get is 100 points. Think of this as partial credit.

Given the extra points, manage your time wisely!

Disclaimer: All data in the first database is taken from LegiScan website. I have made some minor modifications to limit the possibility of hard coding answers. For example, Jesse Helms, a deceased NC legislator, and James K. Polk 11th President of the United States and UNC alumnus, have been added to one of the databases used.

None of these changes are made to present a political bias. Any errors in the transcription of the data from LegiScan to the exam database are mine. If you see any errors, please let me know so that I may make timely corrections.

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 exam 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 **MAY 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.

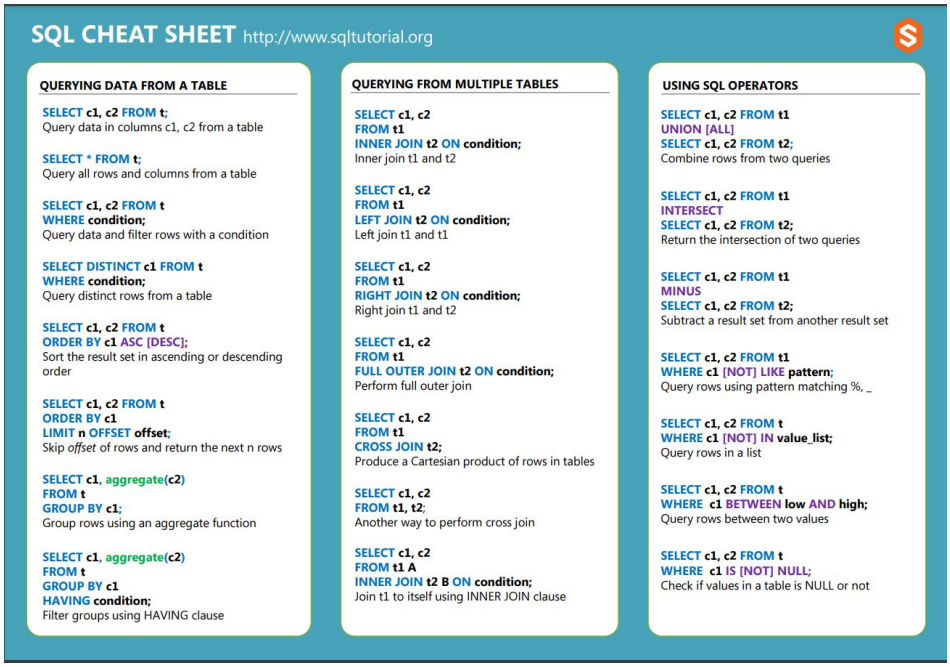
Note that HTML select statements with drop-downs are simple multiple choice questions. No highlighting of correct answers are done for select questions.

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

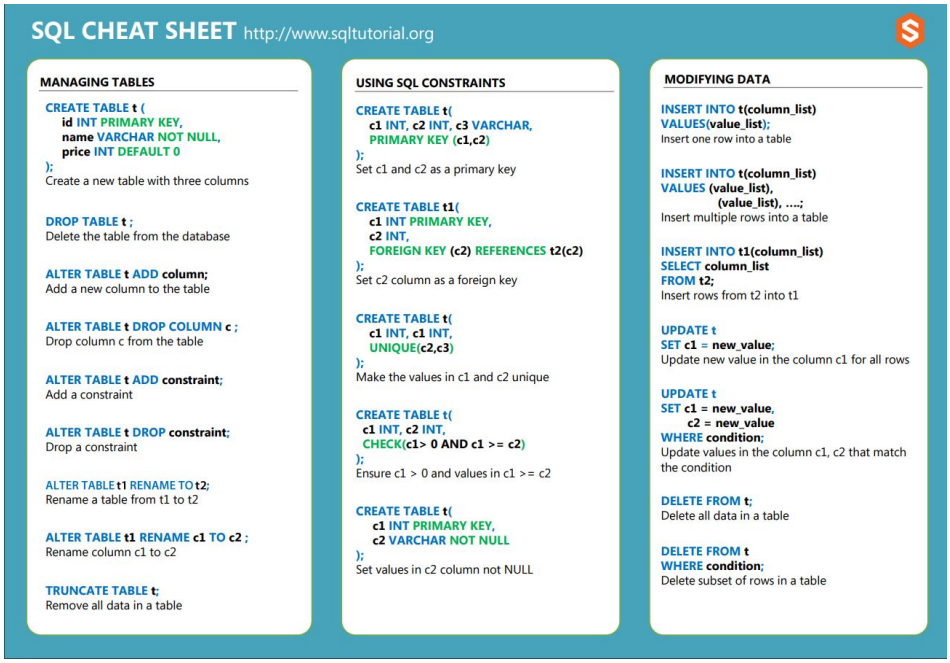
```
SELECT c1, c2 FROM t1
MINUS
SELECT c1, c2 FROM t2;
```
Subtract a result set from another result set

```
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

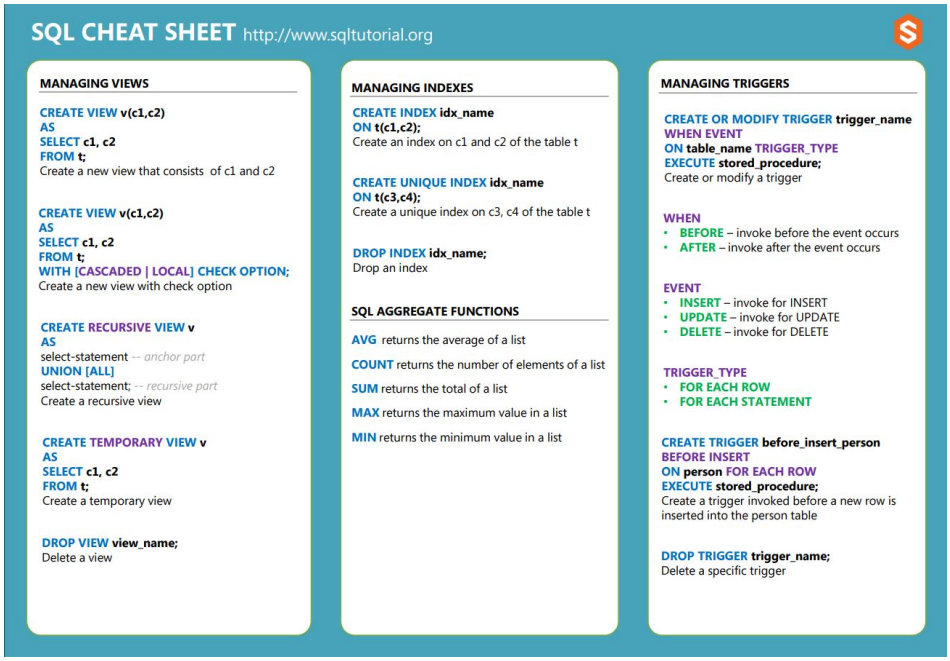
```
INSERT INTO t1(column_list)
SELECT column_list
FROM t2;
```
Insert rows from t2 into t1

```
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



Sailor Database Schema

Here are the tables you'll find for the 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, -- yyyy-mm-dd
     FOREIGN KEY (sid) REFERENCES Sailors(sid)
     ON DELETE CASCADE
     FOREIGN KEY (bid) REFERENCES Boats(bid)
     ON DELETE CASCADE)
```

Legislature Database Schema

Here are the tables you'll find for the 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 Legislators
     (legislator_id INTEGER PRIMARY KEY,
     party TEXT CHECK(party IN (`Democratic`, `Republicar
     role text CHECK(role IN (`Senator`, `Representative`
     name text,
     first_name text,
     middle_name text,
     last_name text,
     district text CHECK(SUBSTR(district, 1,3) IN (`HD-`,
     )
CREATE TABLE Bills
     (bill_id INTEGER PRIMARY KEY,
     title TEXT NOT NULL,
     status TEXT CHECK(status IN (`Introduced`, `Passes`,
     status_date TEXT, -- YYYY-MM-DD
     url TEXT)
CREATE TABLE Roll_calls
     (roll_call_id INTEGER PRIMARY KEY,
     bill_id INTEGER REFERENCES Bills(bill_id) NOT NULL,
     description TEXT,
     date TEXT, -- YYYY-MM-DD if available
     yea INTEGER UNSIGNED, -- Count of Yeas
     nay INTEGER UNSIGNED, -- Count of Nays
     not_voting INTEGER UNSIGNED, -- Count of not voting
     absent INTEGER UNSIGNED, -- Count of absent
     passed INTEGER CHECK(passed IN (0, 1)),
     status_date TEXT, -- YYYY-MM-DD
     chamber TEXT CHECK(chamber IN (`House`, `Senate`)))
CREATE TABLE Votes
     (roll_call_id INTEGER REFERENCES Roll_calls(Roll_cal
     legislator_id INTEGER REFERENCES Legislators(legisla
     vote_text TEXT CHECK(vote_text IN (`Yea`, `Nay`, `Ab
     UNIQUE(roll_call_id, legislator_id))
CREATE TABLE Sponsors
     (bill_id INTEGER REFERENCES Bills(bill_id),
```

```
    legislator_id INTEGER REFERENCES Legislators(legisla
    UNIQUE(bill_id, legislator_id))
CREATE TABLE Subject_names
    (subject_id INTEGER PRIMARY KEY,
    subject_name TEXT,
    UNIQUE(subject_name))
CREATE TABLE Bill_subjects
    (subject_id INTEGER REFERENCES Subject_names(subject
    bill_id INTEGER REFERENCES Bills(bill_id) NOT NULL,
    UNIQUE(subject_id, bill_id))
```

Scratch area

The following scratch space can be used to help develop and test queries against the **first** of the databases described above.

```
Select * from Legislators, Bills, Sponsors
```

Execute | Minimize Output

Questions For a total of 100 points

SQL Queries 45 points

In this section, you will write SQL queries for the North Carolina legislature schema at the beginning of the exam. Your queries will be tested immediately against **two** different databases. The second database has small modifications to prevent hard coding answers. 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.

Partisans.2: List the number of Democratics in the North Carolina Senate.

Execute 5 points Minimize Output

Subjects.Democratic.2: Give an alphabetized list of bill subjects which had a Democratic bill sponsor from the North Carolina Senate.

Execute 10 points Minimize Output

Bill.Most.Sponsors.2: Give the url of the bill(s) in the North Carolina Senate with the most number of Democratic sponsors.

Execute 15 points Minimize Output

Only.One.Party.Votes.Nay.2: Give the url of the bill(s) in the North Carolina Senate in ascending order which had some vote (roll call) where some Democrats voted against (vote_text = Nay) the bill and which had the same vote or some other vote (roll call) for that bill where no non-Democrats voted against the bill (vote_text = Nay).
Note: There are many votes (roll calls). This question is not asking for Nay/Non-nay for a single roll call. That would be much narrower query with fewer numbers of bills output.

Execute 15 points Minimize Output

B+ Trees 35 points

Figure 1 is a B+ Tree with index pages that can contain at most four keys and leaf pages that can contain at most four data entries.

Assume that the tree was initially empty and that the key values of 11, 12, 13, 14, 15, 16, 17, 18, and 19 are inserted in this order into the B+ tree. Use the table at the right to fill in the root's keys and the leaf data values for the B+ tree after the insertions. 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 any values that are empty, enter `null`.

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

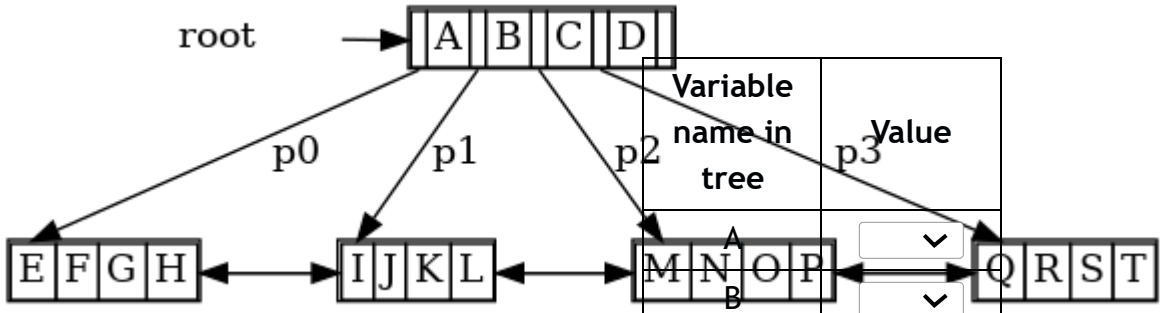


Figure 1. Initially empty with insertions into a B+ tree

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

N	
O	
P	
Q	
R	
S	
T	

15 points

Figure 2 is a B+ Tree with index pages that can contain at most four keys and leaf pages that can contain at most four data entries. The layout is similar to Figure 1, but the actual values may or may not be identical to Figure 1.

For this question, assume you will bulk load the tree with the values 11, 12, 13, 14, 15, 16, 17, 18, and 19. Use the table at the right to fill in the root's keys and the leaf data values for the B+ tree after the insertions. 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 any values that are empty, enter `null`.

Btree.Bulk.Load.2: Fill in the key and data values for the resulting B+ tree after bulk loading.
Note: Even though, alternative 2 leaf pages include an asterisk in the notation, the pull downs do not include the asterisk.

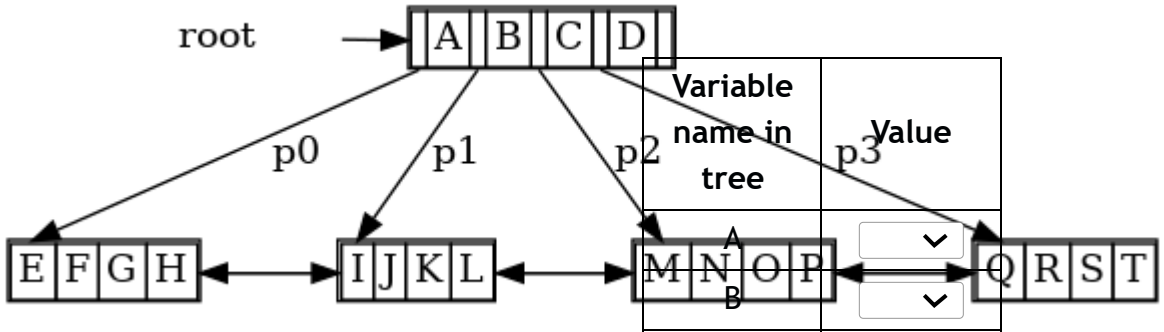


Figure 2. Bulk loaded B+ tree

Variable name in tree	Value
A	
B	
C	
D	
E	

F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	

10 points

Btree.Valid.2: For the following B+ trees, select Valid from the pull down if the B+ tree is valid. Otherwise select the **single node** (root, leaf 1, etc) that is invalid within the B+ tree and must be changed to make the tree valid. Therefore, if the tree is invalid and there is one single node that has to be changed to fix it, select that node.

Note: In an effort to fit the images horizontally, the double arrow linking leaf nodes may appear to be a diamond. But it is a compressed doubly linked list arrow.

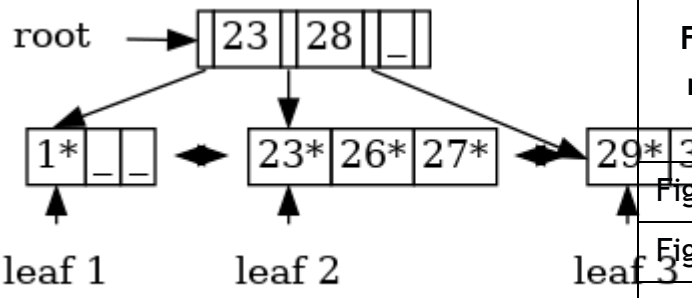


Figure name	Valid or index that's invalid
Figure 4.1	
Figure 4.2	
Figure 4.3	

Figure 4.1

Figure 4.4

10 points

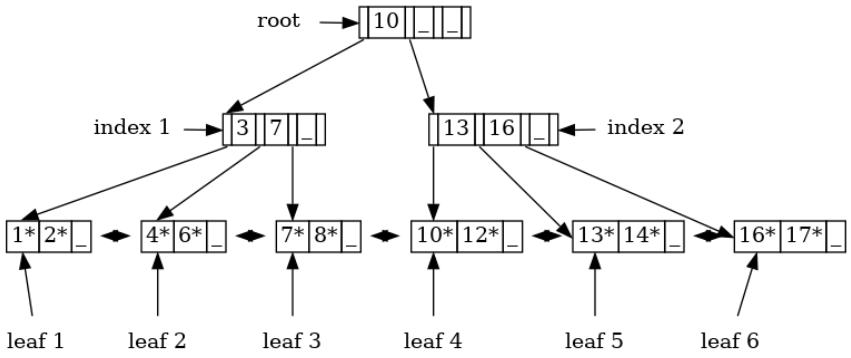


Figure 4.2

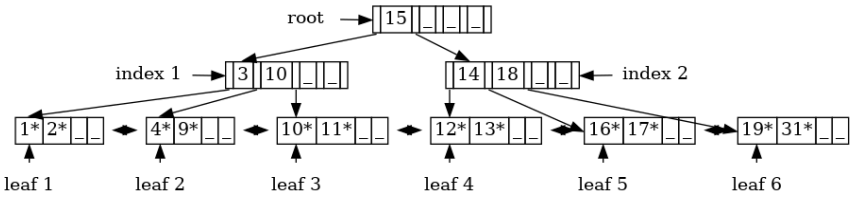


Figure 4.3

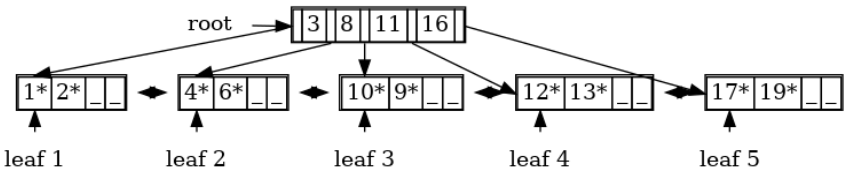


Figure 4.4

I/O Evaluations 30 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: For the following evaluation questions assume that **no** buffering is used as we haven't covered that in class yet. **Also note that to facilitate the math, the hash I/O require, on average 2 I/Os** not the normal 1.2 I/Os as discussed in class.

Assume that each tuple of Sailors is 250 bytes long, that on average with 67% occupancy each page holds 10 Sailors tuples, and there are 20 pages. Sailor integer ratings are evenly distributed from 1 to 10, inclusive.

There is a hash index on relations Sailors $\langle$ Sailors.rating, Sailors.age $\rangle$ and on average the hash require 2 I/Os.

Sailor.Evaluation.2: How many I/Os are required to evaluate $\sigma_{rating=6}Sailors$?

I/Os required	Answer
If the data entries are clustered alternative 1	
If the data entries are unclustered alternative 2	

10 points

Further assume that each tuple of Reserves is 500 bytes long, that on average with 67% occupancy each page holds 5 Reserves tuples, and there are 20 pages. Reserves integer sailor ids are evenly distributed from 1 to 10, inclusive.

There is a hash index on relations Reserves $\langle$ Reserves.sid $\rangle$ and on average the hash requires 2 I/Os.

Simple.Nested.Natural.Join.Evaluation.2: Assuming both relations have clustered data entries, how many I/Os are required to evaluate $Sailors \bowtie Reserves$?

I/Os required	Answer
If a simple nested loop with Sailors as the outer loop is used	
If a simple nested loop with Reserves as the outer loop is used	

10 points

Index.Nested.Natural.Join.Evaluation.2: Assuming both relations have clustered data entries, how many I/Os are required to evaluate $Sailors \bowtie Reserves$?

I/Os required	Answer
If an index nested loop with Sailors as the outer loop is used	
If an index nested loop with Reserves as the outer loop is used	

10 points
