

Preview Test: INFS7901 Semester One Final Examination 2021

Test Information

Description Undertaking this online examination deems your commitment to UQ's academic integrity pledge as summarised in the following declaration:

"I certify that I have completed this examination in an honest, fair and trustworthy manner, that my submitted answers are entirely my own work, and that I have neither given nor received any unauthorised assistance on this examination".

Instructions You need to answer all of the questions in the Blackboard Test.

Timed Test This test has a time limit of 2 hours and 10 minutes. This test will save and be submitted automatically when the time expires.

Warnings appear when **half the time, 5 minutes, 1 minute**, and **30 seconds** remain.
[The timer does not appear when previewing this test]

Multiple Attempts Not allowed. This test can only be taken once.

Force Completion This test can be saved and resumed at any point until the time has expired. The timer will continue to run if you leave the test.

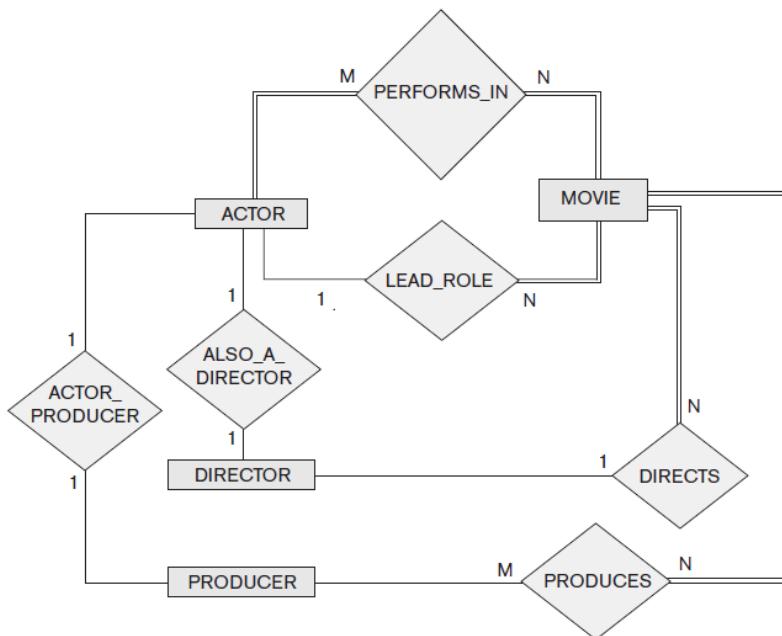
Your answers are saved automatically.

QUESTION 1

8 points

[Save Answer](#)

Consider the following ER schema for a MOVIES database. Assume that MOVIES is a populated database. ACTOR is used as a generic term and includes actresses. Given the constraints shown in the ER schema, respond to the following statements with True or False..



- There are no actors in the database that have been in no movies. A. True
B. False *True*
- Some actors might be producers that have not produced any movie. *True*
- All the leading roles are played by actors who are also directors. *False*
- There is no movie without a leading role. *True*
- A producer can be both a director and an actor. *True*
- There can be movies with directors and producers but with no actors. *False*
- A movie must have multiple producers. *False*
- There can be a movie in which multiple actors play that are also directors in other movies. *True*

QUESTION 2

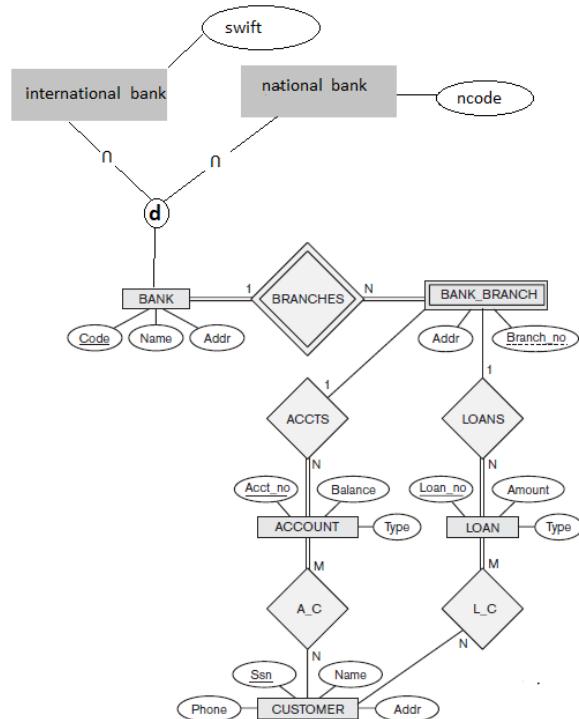
6 points

[Save Answer](#)

Consider the following schema. Map

- **Bank_BRANCH**, [2 marks]
- **LOANS** (the relation, not the entity LOAN), and [2 marks]
- **international bank** [2 marks]

into their corresponding schema e.g. Table1(A, B, C) B references Table2.B.



Bank Branch (Code, Branch_no, addr)

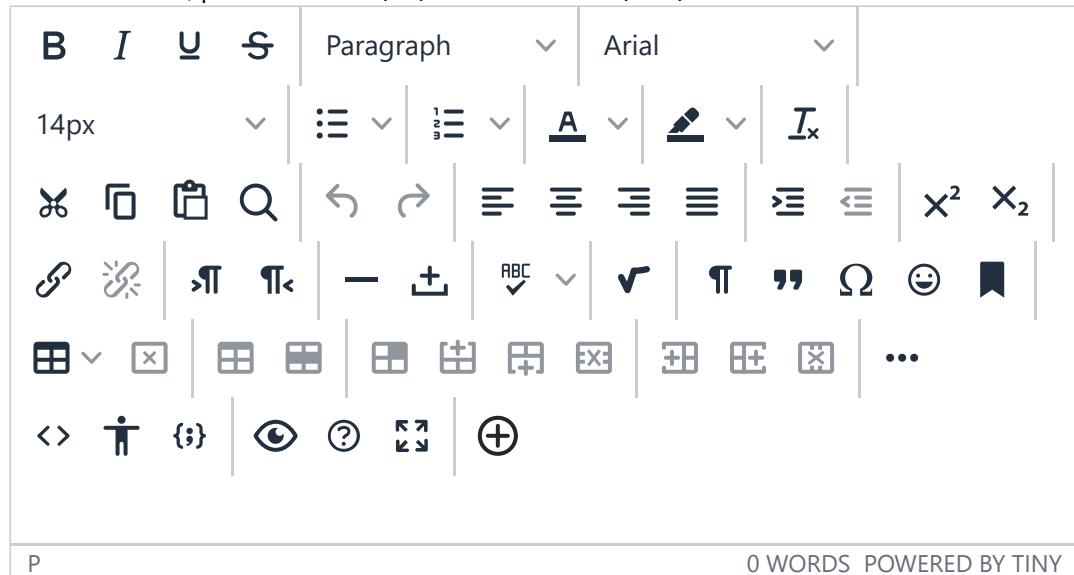
Bank Branch. Code → Bank. Code.

International_Bank (Code, Swift)

International_Bank. Code → Bank. Code

Loans(Loan_no, Code, Branch_no)

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).



QUESTION 3**4 points**

Save Answer

In relation R(ABCDEF) with the following functional dependencies (FDs)

- A → B
- B → C
- D → E
- F → D

a) What are the candidate key(s)? $\{A, \bar{F}\}$

b) Is R BCNF? why or why not? No, because $D \rightarrow E, \bar{F} \rightarrow D$ violating BCNF

c) Which of the four FDs should be removed so that R becomes 3NF?

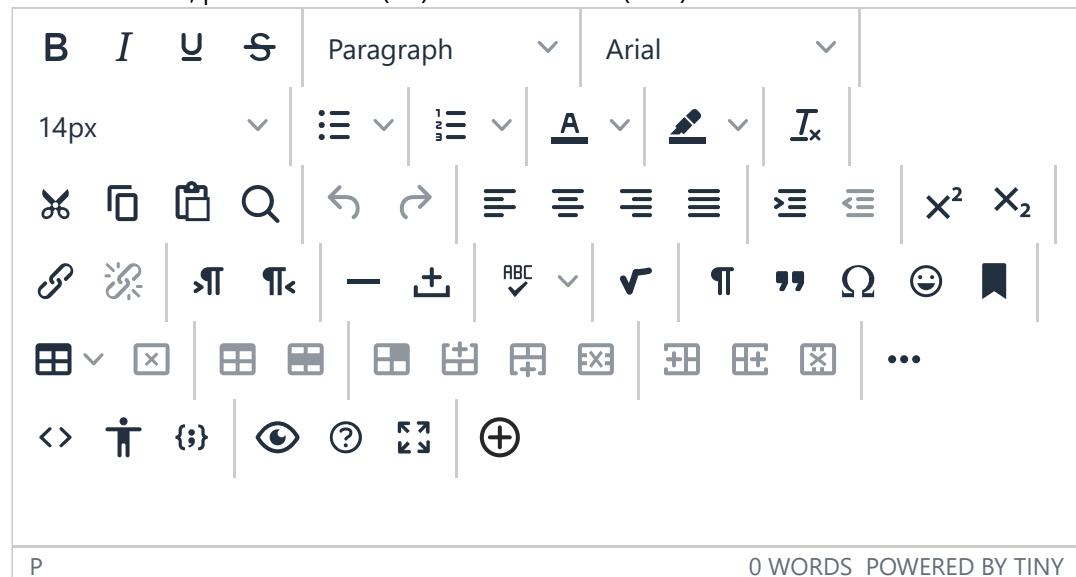
d) what is the closure of {AD}+?

$$(AD)^+ = ABC\bar{E}$$

$$\begin{array}{l} D \rightarrow E \\ F \rightarrow D \end{array}$$

Each 1 mark.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).



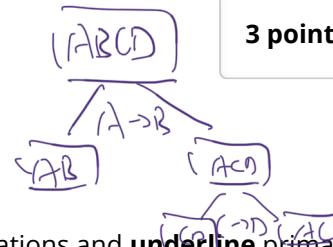
QUESTION 4

3 points

Save Answer

Given R(ABCD) with functional dependencies:

- A → B
- C → D



$$R_1 = \underline{A} B$$

$$R_2 = \underline{C} D$$

$$R_3 = \underline{A} C$$

decompose R and write the **final** resulting relations and **underline** primary keys.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

P 0 WORDS POWERED BY TINY

QUESTION 5**7 points**

Save Answer

Consider the following relational schema, in which the students attend an online data science course website that offers courses in three levels of beginner, intermediate and advanced. The website then issues certificates for each course that a student passes. If a student does not pass the course, it won't be stored in the database.

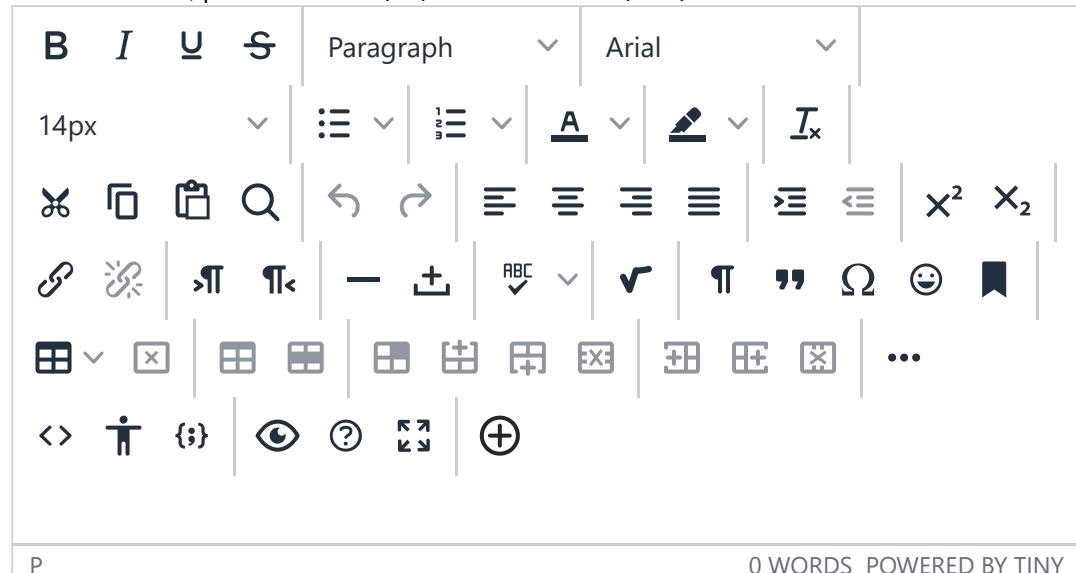
- b. Select Student.sid
from Student L
join CertificateC on
c.sid = C.cid
where C.level is null
- Student (sid, name, address)
 - Course (cid, title, level)
 - Certificate (sid, cid, date)
 - Certificate.sid references Student.sid
 - Certificate.cid references Course.cid

Course level can be basic, intermediate or advanced.

Write the following queries in SQL:

- a. List the id of students whose name starts with letter A. [1 mark]
- b. List the id of students without any certificate. [1 mark]
- c. List level and the total number of certificates issued for all the courses in that level. [1 mark]
Select Level, Count(Cid) from Course. Group by Level;
- d. List id and name of students that have certificates in all three beginner, intermediate and advanced levels (you can assume division is supported). [2 marks]
- e. List pair of course names, and the number of students that have passed both. [2 marks]

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).



QUESTION 6**2 points**

Save Answer

What is the worst-case complexity of function3(A) in Python language where A is an array with length n assuming that the complexity of function2 is O(n)?

function3(A):

if function2(A) == True:

merge_sort(A)

$n^2 \log n$

C

- a. $O(n \log n)$
- b. $O(n)$
- c. $O(n^2 \log n)$
- d. $O(n^2)$

QUESTION 7**4 points**

Save Answer

For each asymptotic comparison select if it is True or False:

- T(A) = $n^2 + 16\log(n)$ is A. False
faster than T(B) = $2n^2 + n^3$ B. True

True

- T(A) = $n^2 / \log(n)$ is True
faster than T(B) = $\log(n^3)$

True

- T(A) = $n \log(n)$ is faster False
than T(B) = $4^{\log(n)} + n$

False

- T(A) = $n \log(n)$ is True
faster than T(B) = $n! / (n-2)!$

True

QUESTION 8**2 points**

Save Answer

A student has designed a small DBMS to be deployed on low-memory mobile phones. Which of the following sorting algorithms is appropriate for "order by" operations on multiple columns in the implemented SQL?

- selection sort
- insertion sort
- quick sort
- merge sort

D

QUESTION 9**2 points**

Save Answer

Match array [9 6 8 2 10 4] in its fourth iteration of sorting with the algorithm that **could be** sorting it.

- ▾ 2 6 8 9 10 4 B
- ▾ 2 4 6 9 10 8 A

- A. selection sort
- B. insertion sort

QUESTION 10**1 points**

Save Answer

After the first iteration of QuickSort the array is **3 9 8 2**. Which item could have been selected as pivot?

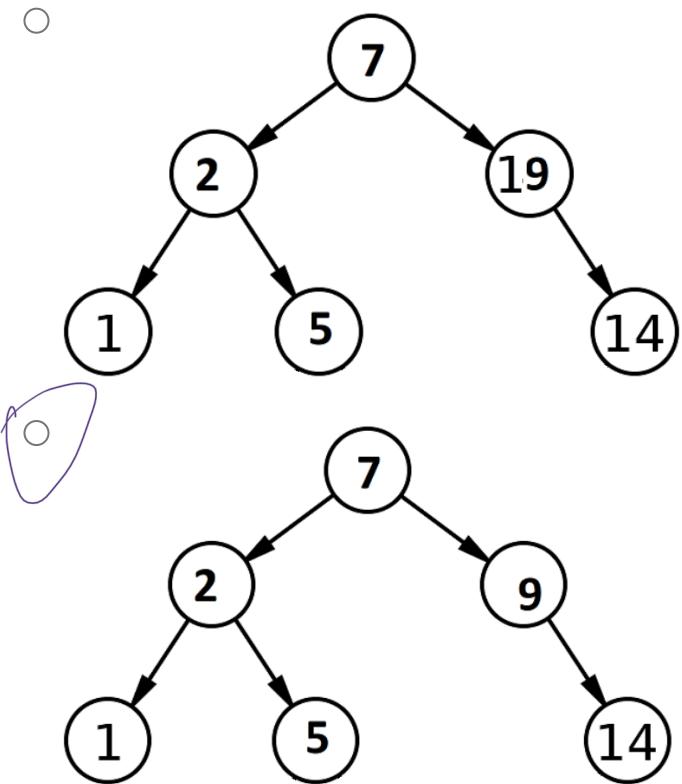
- 3
- 9
- 8
- 2

3

QUESTION 11**1 points**

Save Answer

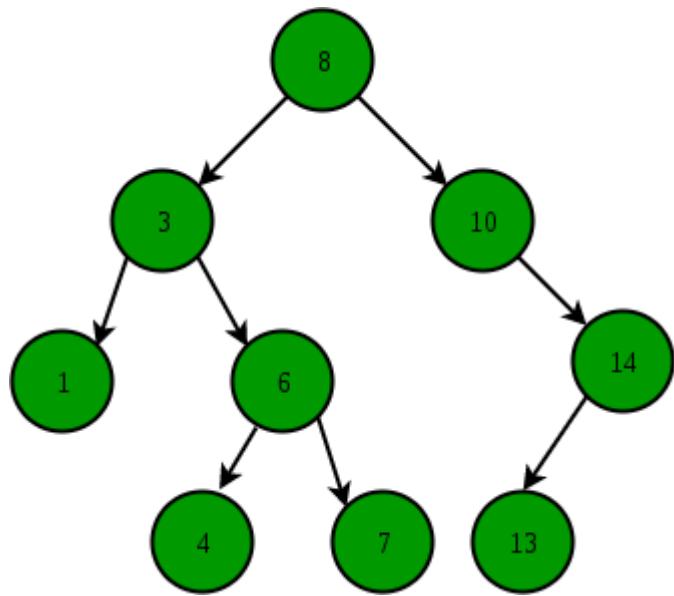
Which one is a binary search tree?



QUESTION 12**2 points**

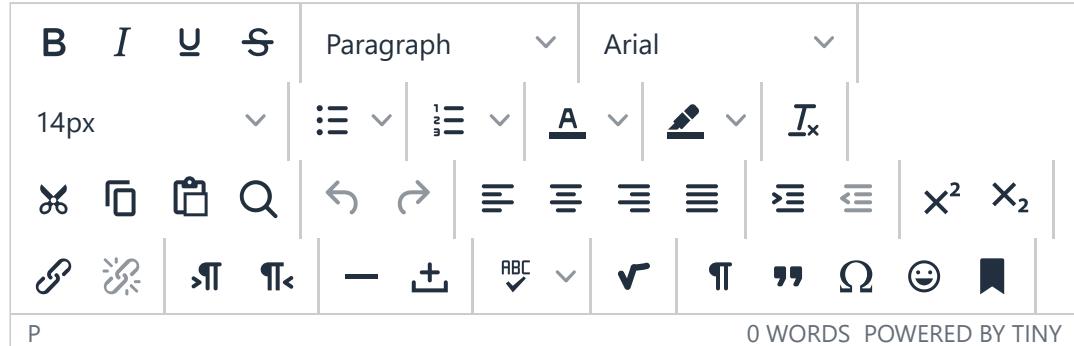
Save Answer

List the nodes (in order) that will be visited if we search for **5** in the following binary search tree (BST)?



8, 3, 6, 4

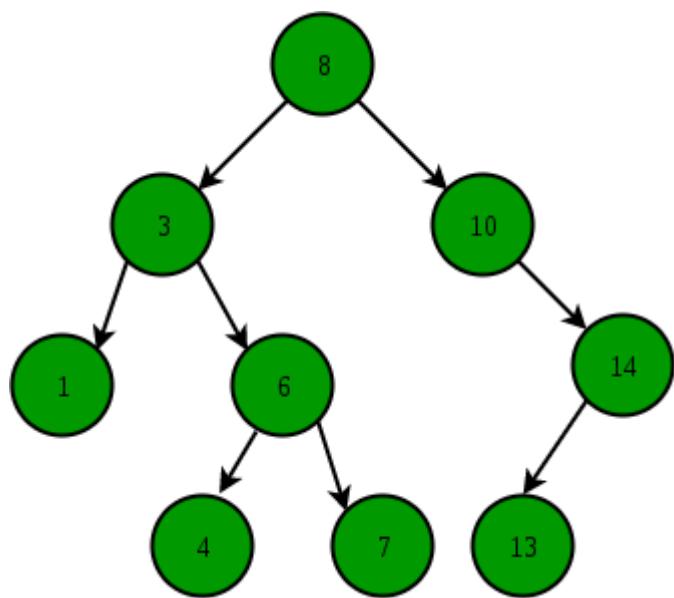
For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).



QUESTION 13**3 points**

Save Answer

In the following binary search tree, we first **delete 3**. What will be the nodes that will be visited (in order) if now we **search** for **5**?



8. 3. 6. 4.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

A toolbar interface for a rich text editor. It includes sections for bold (B), italic (I), underline (U), strikethrough (S), Paragraph style (with dropdowns for alignment and font size), Arial font, and a list of icons for various document functions like linking, tables, and symbols.

The toolbar also includes a text input field containing the letter 'P' and a status bar indicating '0 WORDS POWERED BY TINY'.

QUESTION 14**5 points**

Save Answer

A hashtable with size 5 uses open addressing with quadratic probing.

If we insert:

15, 16, 25, 31 which values will be in indices of the hashtable (0, 1, 2, 3, 4)?

- 0 15
- 1 16
- 2 16
- 3 empty
- 4 25

- A. empty
- B. 15
- C. 16
- D. 31
- E. 25

$$h(i) = \begin{cases} 0 & i=0 \\ 1 & i=1 \\ 2 & i=2 \\ 3 & i=3 \\ 4 & i=4 \end{cases}$$

QUESTION 15**2 points**

Save Answer

In open addressing with **linear probing**, will we face a situation where there exists an empty index, but we can't insert a value into the hashtable?

- Yes
- No

Yes

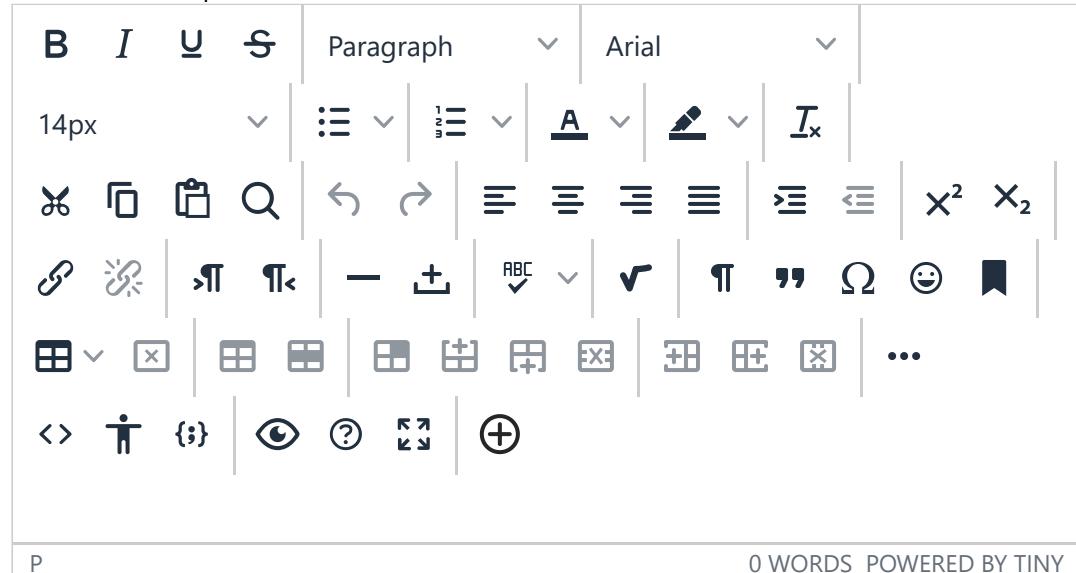
QUESTION 16**3 points**

Save Answer

In a hashtable of size 10 with linked-list chaining, we have inserted 100 values.

- a) what is the load factor? [1 mark] $= \frac{100}{10} = 10$
- b) what is the worst case search complexity of such a hashtable? [1 mark] $O(100)$
- c) If all the values that we insert are multiples of 10, what is the complexity of searching for numbers which are **not** multiples of 10? [1 mark] $O(n)$

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).



QUESTION 17**6 points**

Save Answer

Answer the following questions about indexing:

Given the schema below what kind of index will you design for each of the four frequent queries below?

- Customer(cid, cname, age)
- Product(pid, pname)
- Purchase(cid, pid, date, price)
 - Purchase.cid references Customer.cid
 - Purchase.pid references Product.pid

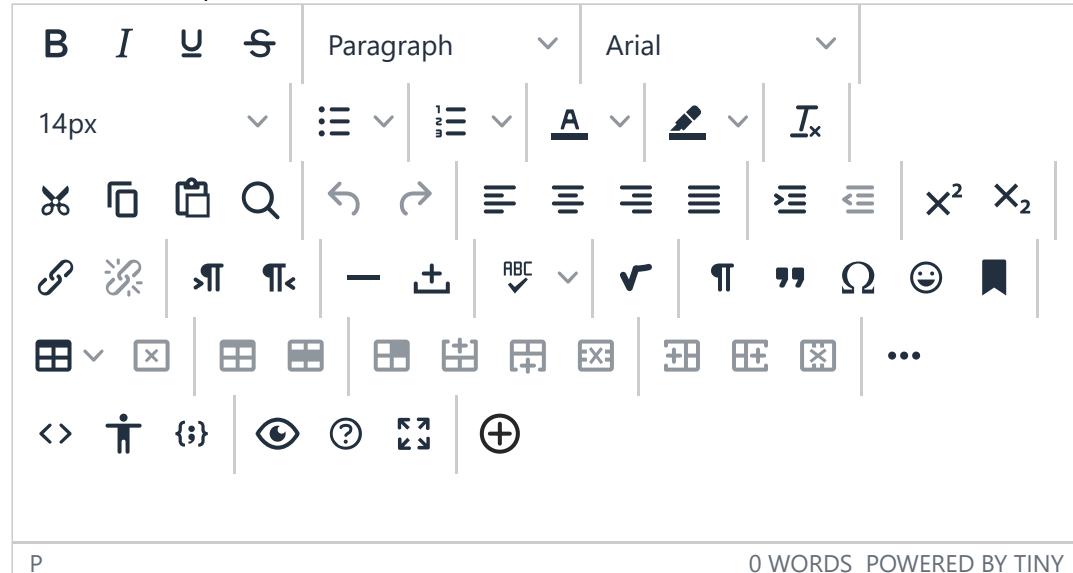
For each query, mention if the index

- is primary or secondary?
- uses tree or hash?
- is dense or sparse?

e.g. primary, using tree, sparse

- ① Secondary, hash. Dense
- ② Primary, tree. Sparse
- ③ Secondary, hash. Dense
- ④ Primary, tree. Sparse
1. (1.5 marks) select * from Customer where cname="John";
 2. (1.5 marks) select * from Purchase where cid="Jane" and pid="P127";
 3. (1.5 marks) select * from Customer where age > 80;
 4. (1.5 marks) select * from Product where pid="P127";

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).



QUESTION 18**1 points**

Save Answer

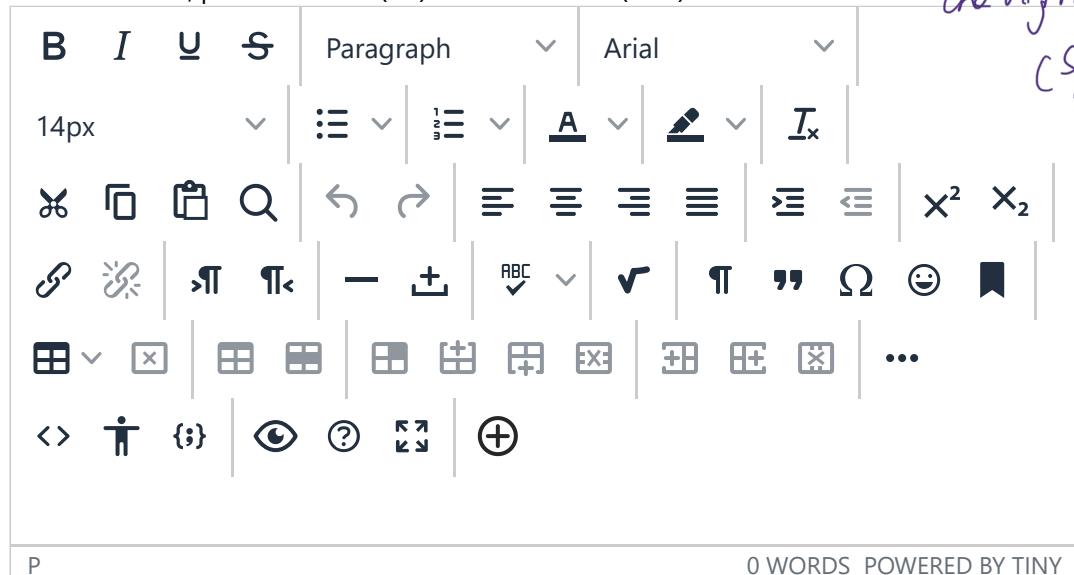
Consider schema STUDENT(sid, name, is_adult) and a very frequent query "select * from STUDENT where is_adult=True". We know that more than 98% of students are adults (is_adult=True).

NO, because it would not significantly improve the query

Do you recommend creating a secondary (non-key) index for is_adult field on this relation? please support your yes/no answer in one sentence.

Performance due to
the high selectivity
(98%) of the
is_adult=true
condition

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).



QUESTION 19

8 points

[Save Answer](#)

Consider the following relational schema, in which the Purchase lists the products that customers buy.

- Customer(cid, cname, age)
- Product(pid, pname)
- Purchase(cid, pid, date, price)
 - Purchase.cid references Customer.cid
 - Purchase.pid references Product.pid

$\emptyset \cap_{cid} (\delta_{age > 50} Customer)$

Write the following queries in relational algebra

$\emptyset \cap_{cname} (R_1 \times R_2)$

For symbols you can either copy symbols from below or use their name

- Pi π
- Sigma σ
- Join \bowtie
- Cartesian Product X

$R_1 = \delta_{Product.pid = Purchase.pid \text{ and } Pname = "TV"}$

$R_2 = \delta_{Customer.cid : Purchase.cid \text{ and } age > 30}$ (Product \bowtie Purchase)

a) (1 mark) Find the id of all customers who are older than 50.

b) (2 marks) Find the name and age of all customers who are older than 30 and have purchased a product with pname="TV".

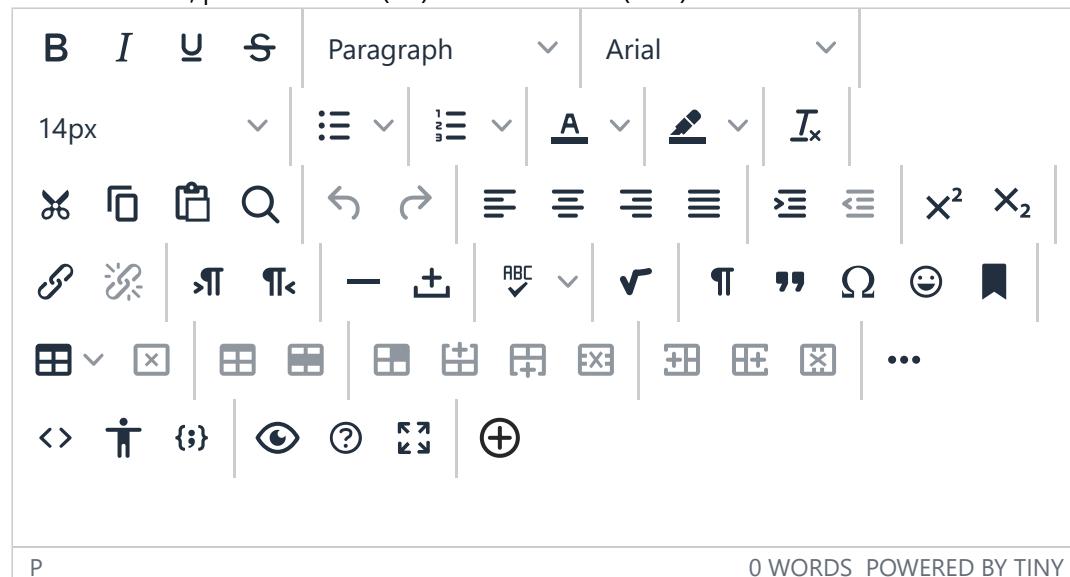
c) (5 marks) Write the initial and optimised query plan for query

"select customer.cname from customer, product, purchase where customer.age > 70 and product.pname='TV' and purchase.price > 1000 and product.pid=purchase.pid and customer.cid=purchase.cid" that lists names of older customers who have bought rather expensive TVs.

initial (1.5 marks)

optimised (3.5 marks)

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).



QUESTION 20**0 points**

Save Answer

Please use this space to specify any assumptions you have made in completing the exam and which questions those assumptions relate to. You may also include queries you may have made with respect to a particular question, should you have been able to 'raise your hand' in an examination room.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

