Preview Test: INFS7901 Semester One Final Examination 2020

Test Information

Description	INFS7901 Database Principles Semester One 2020 - Final Examination This is an open book exam – all materials permitted.
Instructions	Answer all questions. There are **70** marks in total. If you experience a technical error during the exam, you should do the following: • Contact the Library AskUs service for advice. • Request an email from AskUs documenting the advice provided so you can forward it to your course coordinator. • Inform the course coordinator at a.rahimi@uq.edu.au.
Timed Test	This test has a time limit of 2 hours and 30 minutes.This test will save and submit 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 time has expired. The timer will continue to run if you leave the test.

QUESTION 1

0 points Save Answer

This is an open book exam. You will have access to your own notes, course texts and other materials. The normal academic integrity rules apply.

- You cannot cut-and-paste material other than your own work as answers.
- You are not permitted to consult any other person, whether directly, online, or through any other means about any aspect of this assessment during the period that this assessment is available.

If it is found that you have given or sought outside assistance with this assessment then that will be deemed to be cheating and will result in disciplinary action. By undertaking this online assessment you are deemed to have acknowledged UQ's Academic Integrity Pledge and to have made the following declaration:

"I certify that my submitted answers are entirely my own work and that I have neither given nor received any unauthorised assistance on this assessment item".

- ◯ Agree
- ◯ Disagree

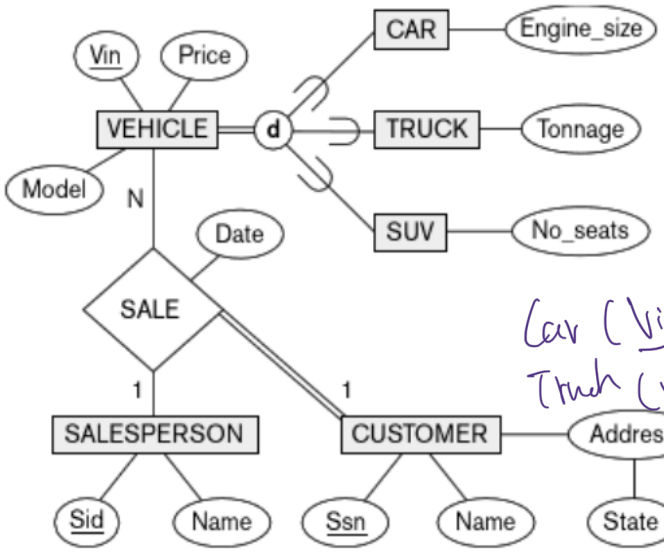
QUESTION 2

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.

QUESTION 3 **8 points** Save Answer

Answer the following questions based on the ER diagram below.



Salesperson (Sid, name)

Customer (Ssn, name, city, street, state, sid)

Vehicle (Vin, price, model, sid, ssn)
date

Car (Vin, Engin_size)

Truck (Vin, Tonnage) SUV (Vin, No_seats)

a. (1 mark) Can we have multiple salespersons in table SALESPERSON (Y/N)? Explain in one sentence. Y. because different Sid represent different salesperson

b. (1 mark) Can we have customer records in table CUSTOMER that have not purchased a vehicle (Y/N)? Explain in one sentence. NO. Customer to Sale is total dependency.

c. (1 mark) Is it possible for a vehicle to have both attributes Engine_size and Tonnage (Y/N)? Explain in one sentence. NO. because vehicle's subclass is disjoint.

d. (5 marks) Map this ER diagram into a relation schema. Make sure to underline attributes of primary key, and make foreign keys bold as in Table(att1, att2, att3, **att4**) where att1 and att2 are primary keys and att4 is a foreign key. If an attribute is both a primary key and a foreign key both underline it and use a bold font (**att**).

QUESTION 4

6 points Save Answer

Consider the relation instance on the right-hand side and answer the following questions:

A	B	C	D
10	b1	c1	1
10	b2	c2	1
11	b4	c1	2
12	b3	c4	3
13	b1	c1	4
14	b3	c4	5

a. (1 marks) Check to see if all of the following dependencies hold with respect to the instance or not:

A → B ✗
A → D ✓
C → B ✗
B → A ✗

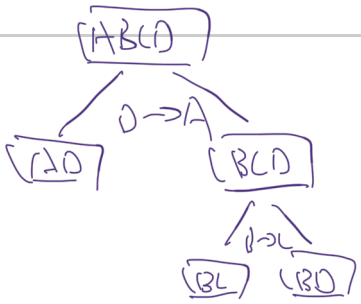
{DB}.

b. (1 marks) Assuming that D → A and B → C hold in the given schema, find a minimal key for the relation.

A⁺ = {A}

c. (1 mark) Assuming that D → A and B → C hold in the given schema, what is the closure of A?

d. (3 marks) Assuming that D → A and B → C hold in the given schema, decompose it, if necessary, so that all the resultant relations are in BCNF (only write the resulting table schemas).



R1. AD
R2. BC
R3. BD

QUESTION 5 **8 points** Save Answer

Consider the following relational schema, in which the students attend an online data science course website that offers courses in levels of beginner, intermediate and advanced. The website then issues certificates for each course that a student pass.

- Student (sid, sname, address)
- Course (cid, cname, level)
- Certificate (sid, cid, date)
  - Certificate.sid references Student.sid
  - Certificate.cid references Course.cid

Write the following queries in SQL

a. (1 mark) Find the name of students who have passed more than one advanced course.

b. (2 mark) Find the name of students who have passed more than one advanced course, but have not passed any intermediate course.

c. (2 mark) Find the name of students who have not passed any popular course. A popular course is a course that more than 50 students have passed.

d. (3 marks) Find the sname of students who have passed more than one advanced course, but have not passed any intermediate course.

QUESTION 6

8 points Save Answer

Answer the following questions.

a. (4 marks) For each of the following pairs of runtimes, determine which one is asymptotically **faster** (has lower computational complexity). Note that all log functions have base 2.

I. $T(A) = n^2 + 16^{\log(n)}$ vs. $T(B) = 2n^2 + n^3$

T(A)

II. $T(A) = n / \log(n) = \log(n)$ vs. T(B)

T(B)

III. $T(A) = 2^{\log(n)} + n^2$ vs. T(B) $= 2^n / n^2$

T(A)

IV. $T(A) = n^2 + n!$ vs. T(B) = $\log(n)\ n!$

Same

O(A) = n!

O(B) = n!

b. (4 marks) Determine the running time of *functionA* in terms of big-O notation. Justify your answer.

$\log n \begin{cases} \sum_{i:n}^{i} = \log n \\ \sum_{i=1}^{n} = \log n \end{cases}$

```
def functionA(n):
  i = n
   while i > 1:
     functionB(i)
     i = ⌊i / 2⌋

def functionB(n):
  i = 1
  while (i<n):
      i = i*2
      print i
```

QUESTION 7

8 points Save Answer

Answer the following questions.

a. (2 marks) You have an integer array with size one million that is almost sorted and only few items (k unsorted items where k is much smaller than one million, e.g. k=10) are unsorted. Which of the sorting algorithms (selection sort, insertion sort, mergesort, quicksort) will you use to completely sort the array? Explain your answer in terms of asymptotic complexity.

insertion sort
O(n) is best.

b. (2 marks) Which sorting algorithm will be the worst? Explain your answer in terms of asymptotic complexity.

selection sort
$O(n^2)$

c. (2 marks) What if you the array size was 100 billion? Which sorting algorithm you will use? Why?

merge sort.
$O(n \log n)$
it follows divide and conquer approach where the array is divided into smaller subarrays. sorted recursively. and then merged back together

d. (2 marks) After the first iteration of quick sort the array is partitioned as **[4, 3, 10, 5, 6, 1, 2, 11]**. Which number(s) could have been chosen as pivot?

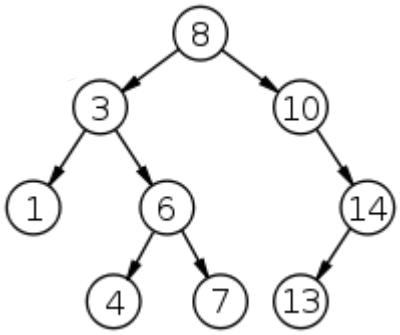
2.

QUESTION 8

8 points Save Answer

Answer the following questions.

This question makes reference to the following Binary Search Tree:



a. (1 mark) Is this tree balanced? Why? NO

b. (1 mark) What are the successor and the predecessor of the root node? 10 4

c. (1 mark) What are the nodes you'll visit if you search for next largest node? 8->10-14

d. (2 mark) What are the nodes you'll visit if you search for 5? 8->3->6-4

e. (3 marks) Delete 8, then search for 5, what are the nodes you'll visit when you search for 5 (5 doesn't exist in the tree)? 10-3-6-4

QUESTION 9

8 points Save Answer

Answer the following questions.

0	
1	
2	
3	
4	
5	
6	

write 8 : index of 1
write 15 : index of 2
write 22 : index of 3
write 6 : index of 5

a. (4 marks) A hash table of size 7 uses open addressing with hash function h(k)=k mod 7, and linear probing. After you insert values [8, 15, 22, 6], what will be the indices of these values? E.g. write 8:indexof8, 15: indexof15 22:indexof22, 4:indexof6.

b. (1 mark) After inserting the values in part *a*, what will be the load factor of the hash table? $\frac{4}{7}$

c. (1 mark) After inserting the values in part *a*, how many more values do we need to insert so that the load factor becomes 1? 3

d. (2 marks) We have an array where values are all powers of two 2, 4, 8, 16, …., etc. If we use linked-list chaining and hash function h(k) = k % 2, After inserting n numbers, what will be the complexity of searching for a number which is a power of two? What will be the complexity of searching for an odd number such as 3? O(1)

O(n)

QUESTION 10 **8 points** Save Answer

Answer the following questions about indexing:

Given schema below what kind of index will you design (primary or secondary, B+ tree or hash-based) 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

a. (2 mark) select * from Customer where cname="something"; secondary hash-based

b. (2 mark) select * from Purchase where cid="something" and pid="somethingelse"; primary grid file

c. (2 marks) select * from Customer where age > 80; secondary hash-based

d. (2 marks) select * from Product where pid="something"; primary B+ tree

QUESTION 11 **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

a. Write the following queries in relational algebra
    I. (2 marks) Find the name of all customers whose age is more than 90.

    II. (2 marks) Find the name and age of all customers that purchased a product with pname=”phone” on date=2020/01/01.

b. (4 marks) Write the relational algebra of the **optimised** query plan for the following query: find the name and age of all customers that purchased a product with pname=”phone” on date=2020/01/01.