Computer Science 3319a

Midterm Exam – Two Hours

Fall, 2014

Name: _____ **Western User ID:** _____

Questions 1 – 6: (12 Marks): _____
Question 7: (23 Marks): _____
Question 8: (42 Marks): _____
Question 9: (30 Marks): _____

Total: _____ **(Out of 137)**

Final Mark: _____ (% Out of 100)

Questions 1-6 (2 Marks each): **PICK THE BEST ANSWER:**

1) Suppose we have a table called **A** with the attributes **a, b, c** and **d**, another table called **B** with the attributes **b, c, d, e, f** and **g**, another table **C** with attributes **d, e, s** and **t**. If we do $(A \bowtie B) \times (((\Pi_{d,e} B) - (\Pi_{d,e} C)) \bowtie B)$, how many attributes will our new table have?
   a) 0
   b) 11
   c) 12
   d) 17
   e) none of the above

2) The relationship ***is built from*** in the following situation: *Building* ***is built from*** *Material Type* (*Building* could be house, apartment building, garage, *Material Type* could be wood, bricks, dry wall) is what type of relationship?
   a) One to one
   b) One to many
   c) Many to many

3) The ***insert*** command in SQL is used to: (Question poorly worded so I accepted b or e)
   a) Add extra fields into an existing table
   b) Add extra records into an existing table
   c) Change the current instance of an existing table
   d) It can be used to do all of the above
   e) b and c only

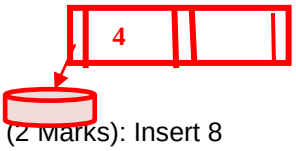
4) Which of the following relationships, contains at least one entity, that must have **total participation** in the relationship.
   a) Woman *marries* Man
   b) Woman *gives birth to* Child
   c) Woman *is sister of* Woman
   d) Woman *is aunt to* Child
   e) None of the above relationships have an entity that must participate totally

5) A dense index is always a primary index, you would never build a dense index on any other field than the primary key.
   a) True
   b) False

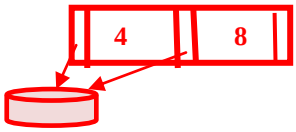
6) Assume you have 2000 records to store. You are going to store them using a heap organization. Each record is 500 bytes long and each block is 2048 bytes. Find the max # of blocks that would have to be accessed to retrieve one record (given the key) in this file.
   a) 1 block
   b) 9 blocks
   c) 250 blocks
   d) 500 blocks
   e) None of the above

7) (23 Marks) Put the following keys into a B+-Tree with a *Pleaf =2 and Porder=3*: 4, 8, 7, 11, 22, 5 and 2 **SHOW ALL YOUR STEPS AND SHOW ALL POINTERS FOR FULL MARKS**

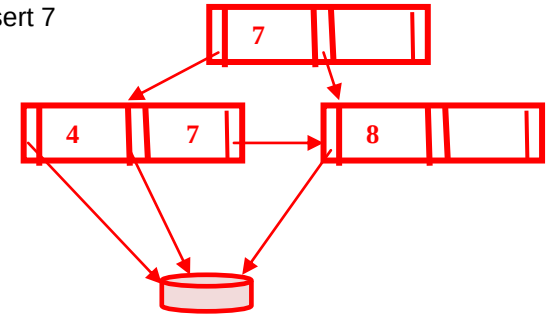
Step A (2 Marks): Insert 4



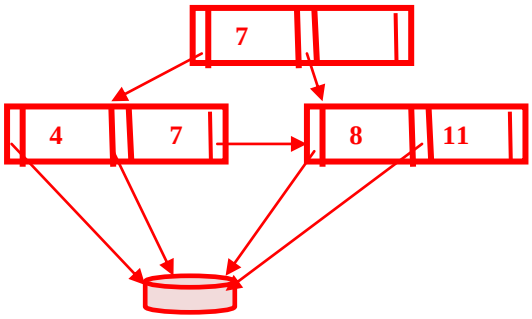
Step B (2 Marks): Insert 8



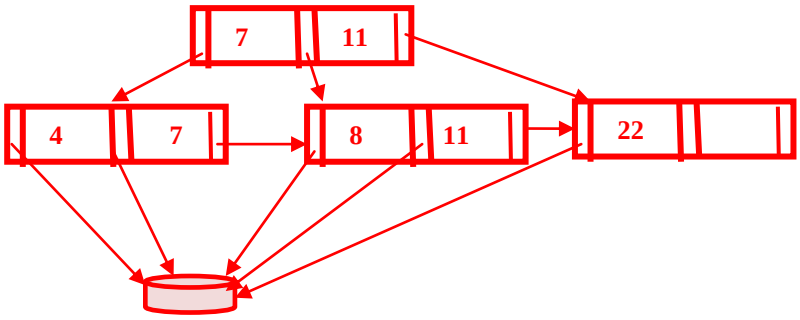
Step C (2 Marks): Insert 7



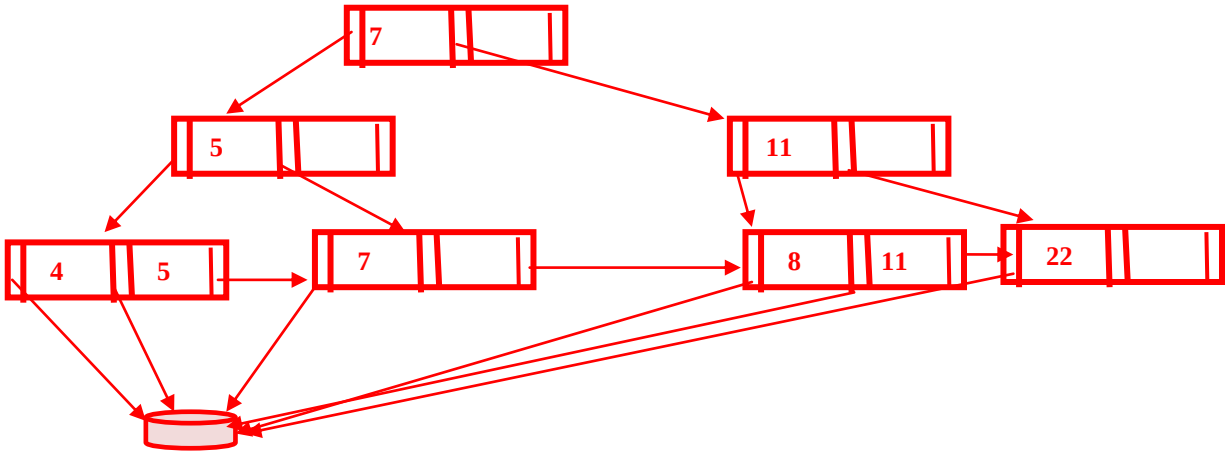
Step D (2 Marks): Insert 11



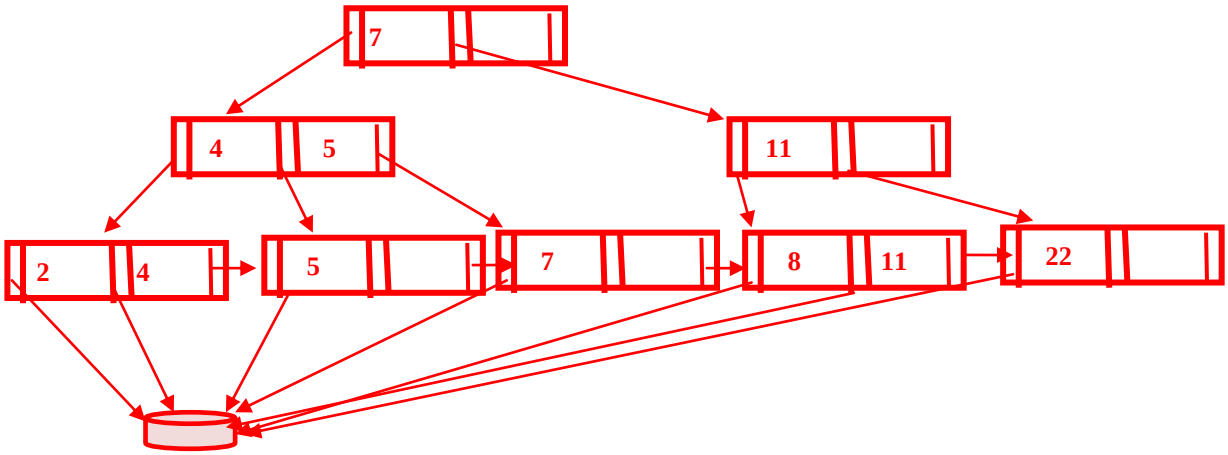
Step E (3 Marks): Insert 22



Step F (5 Marks): Insert 5



Step G (6 Marks): Insert 2



(1 Mark) A B+ tree is a good choice for organizing a table rather than a heap organization for your table if it is a small table and you are doing lots of inserts and very rarely doing searches: TRUE or **FALSE** (circle one)

8. (42 Marks) Consider the following relational schema for a database that keeps track of drivers who lease (i.e. rent) cars.

Write the Relational Algebra expression, the SQL statement and the relational calculus to represent each of the following queries:

DRIVER (DriverLic, FName, LName) e.g. 112, "Homer", "Simpson"
LEASES(DriverLic, LicPlate, StartYearOfLease, kmDriven) e.g. 112, "ALWK 310", 2002, 14000
CAR(LicPlate, Colour, Model, ModelYear) e.g. "ALWK 310", "Blue", "Caravan", 2001

a) Query: Find the Drivers License Number and last name of all drivers whose first name is "Waylan"
   i) (4 Marks) Relational Algebra:

ANSWER ← $\Pi_{DriverLic,LName}$ ($\sigma_{FName="Waylan"}$ DRIVER)

   ii) (4 Marks) SQL:

SELECT DriverLic, LName FROM DRIVER WHERE FName = "Waylan";

   iii) (4 Marks) **Tuple** Relational Calculus:

{d.DriverLic, d.LName | DRIVER(d) and d.FName = "Waylan"}

b) Query: List the first and last name of any driver who leased a car where the car lease started in the same year as the model year of the car
   i) (6 Marks) Relational Algebra

TEMP ← (DRIVER⋈LEASES⋈CAR)

ANSWER ← $\Pi_{FName,LName}$ ($\sigma_{StartYearOfLease=ModelYear}$ TEMP)

   ii) (6 Marks) SQL:

SELECT FName,LName FROM DRIVER, CAR, LEASES WHERE DRIVER.DriverLic=LEASES.DriverLic AND CAR.LicPlate=LEASES.LicPlate and StartYearOfLease=ModelYear;

iii) (6 Marks) Tuple or Domain Relational Calculus:

DRIVER (DriverLic, FName, LName)
LEASES(DriverLic, LicPlate, StartYearOfLease, KMDriven)
CAR(LicPlate, Colour, Model, ModelYear)

{d.FName, d.LName | DRIVER(d) and ((∃s) (∃c) (LEASES(s) and CAR(c) and s.DriverLic=d.DriverLic and c.LicPlate=s.LicPlate and s.StartYearOfLease=cModelYear))}

c) Query: List all the first and last name of any drivers that have never leased a car (Relational Algebra and SQL only)
i) (6 Marks) Relational Algebra:

ALLLeasers ← $\Pi_{DriverLic}$ (LEASES)

ALLDrivers ← $\Pi_{DriverLic}$ (DRIVER)

ANSWER ← $\Pi_{FName,LName}$ (DRIVER⋈ (ALLDrivers – ALLLeasers)

ii) (6 Marks) SQL:

SELECT FName,LName FROM DRIVER WHERE DriverLic NOT IN (SELECT DriverLic FROM LEASES)

9. (30 Marks) Given the following 3 relations **(SHOW any intermediate steps for full marks):**

Table AA:

A	B	C	D
1	2	1	1
1	3	1	3
1	2	2	3
1	4	1	3

Table BB:

E	F	D
1	5	6
2	2	3
5	1	3

Table CC:

A	D	G	H
3	1	1	3
4	2	2	6
5	3	1	6
7	3	1	3

d) (9 Marks) Draw the resulting relation for:

$DD(C,D) \leftarrow \Pi_{G,H}(\sigma_{A<6} CC)$

$GG \leftarrow \Pi_{F,D,C}(DD \bowtie BB)$

$EE(S) \leftarrow \Pi_{D}(AA) - \Pi_{D}(GG)$

$FF \leftarrow (GG \bowtie_{F=S} EE) \bowtie CC$

Table DD

C	D
1	3
2	6
1	6

Table GG

F	D	C
5	6	2
5	6	1
2	3	1
1	3	1

Table EE

S
1

Table FF

F	D	C	S	A	G	H
1	3	1	1	5	1	6
1	3	1	1	7	1	3

e) (5 Marks) Draw the resulting relation for:

$DD(D) \leftarrow \Pi_{B}(\sigma_{B<4}(AA))$

$FF \leftarrow (\Pi_{D,H}(CC)) \div DD$

Table DD

D
2
3

Inbetween Table (before FF)

D	H
1	3
2	6
3	6
3	3

Table FF

H
6

Table AA:

A	B	C	D
1	2	1	1
1	3	1	3
1	2	2	3
1	4	1	3

Table BB:

E	F	D
1	5	6
2	2	3
5	1	3

Table CC:

A	D	G	H
3	1	1	3
4	2	2	6
5	3	1	6
7	3	1	3

f) (4 Marks) Draw the resulting table for the following SQL statement:
SELECT G, D from CC where A > 3 Order by G

G	D
1	3
1	3
2	2

g) (5 Marks) Draw the resulting table for the following SQL statement:
SELECT A, B, BB.D from AA, BB where C = F

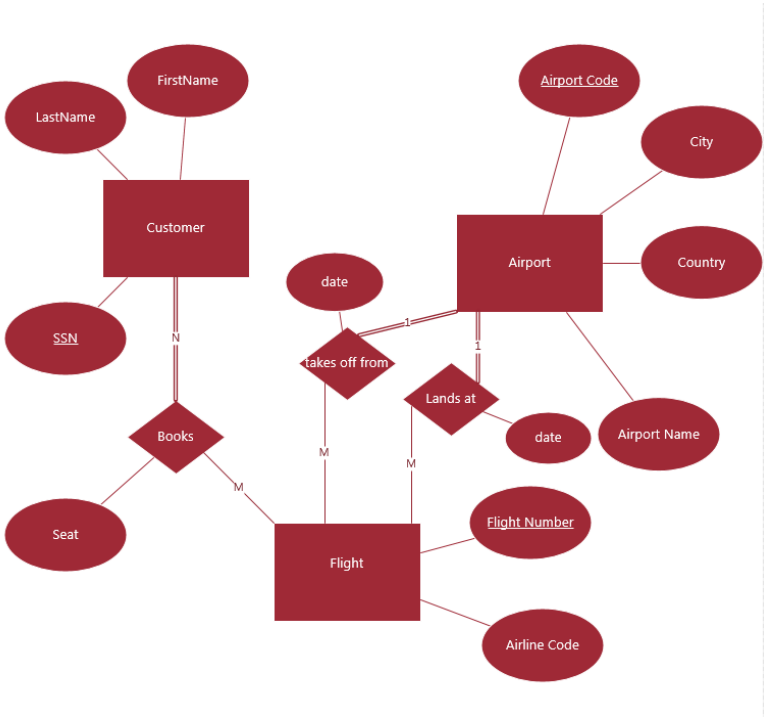
A	B	BB.D
1	2	3
1	3	3
1	2	3
1	4	3

h) (7 Marks) Draw the resulting table for the following SQL statement
SELECT STUFF.B, SUM(JUNK.B) as SUMB FROM AA as STUFF, AA as JUNK where STUFF.B = JUNK.B group by STUFF.B

Stuff.B	SUMB
2	8
3	3
4	4

10. (30 Marks) 2 Parts:

i) (20 Marks) Construct an Entity Relation Diagram for a travel application. Our travel database has customers where each customer has a first and last name and a social insurance number. You must be able to find which flights a customer has booked and which seat the customer booked. Each flight has a flight id number and an airline code. Flights take off from airports. Flights land in airports. A flight can take off from only 1 airport and land in only 1 airport. Airports have airport codes (unique), airport names, city names and country names. We want to keep track of the date the flight took off from an airport and the date the flight landed. Flights can be cancelled so that they don't land in an airport or take off from an airport. A customer must have booked at least one flight to be in our system. No one might book a flight if it is unpopular. Use your common sense to figure out any participation I haven't explicitly stated. DO NOT ADD ANY EXTRA INFORMATION OR MAKE ANY ASSUMPTIONS BECAUSE OF YOUR EXPERIENCES WHEN BOOKING FLIGHTS OR AT AIRPORTS. Label your ER diagram completely. You must show the cardinality and **just use single or double lines to representing participation (NOT (min, max)).**



j) (10 Marks) Map your ER Diagram to the appropriate relational tables. Underline the primary keys. Put a * next to any foreign keys.

Customer

SSN	LastName	FirstName

Airport

AirportCode	City	AirportName	Country

Flight

FlightNumber	AirlineCode	LandsatAirportCode *	TakesOffAirportCode *	LandDate	TakeOffDate

Booking

FlightNumber *	SSN *	SeatNumber

SCRAP PAPER – THIS PAGE INTENTIONALLY LEFT BLANK