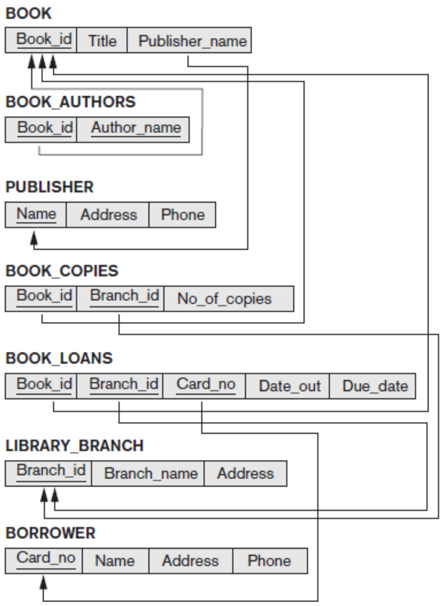
Practice Problems:

Q1.



Consider the LIBRARY relational database schema shown in the figure above, which is used to keep track of books, borrowers, and book loans. Referential integrity constraints are shown as directed arcs in Figure. Write down **relational expressions** for the following queries:

a. How many copies of the book titled *The Lost Tribe* are owned by the library branch whose name is 'Sharpstown'?

b. How many copies of the book titled *The Lost Tribe* are owned by each library branch?

c. Retrieve the names of all borrowers who do not have any books checked out.

d. For each book that is loaned out from the Sharpstown branch and whose Due_date is today, retrieve the book title, the borrower's name, and the borrower's address.

e. For each library branch, retrieve the branch name and the total number of books loaned out from that branch.

f. Retrieve the names, addresses, and number of books checked out for all borrowers who have more than five books checked out.

g. For each book authored (or coauthored) by Stephen King, retrieve the title and the number of copies owned by the library branch whose name is Central.

Q2.

Specify the following queries in **relational algebra** on the database schema given in Exercise 5.14 of your textbook:

a. List the Order# and Ship_date for all orders shipped from Warehouse# W2.

b. List the WAREHOUSE information from which the CUSTOMER named Jose Lopez was supplied his orders. Produce a listing: Order#, Warehouse#.

c. Produce a listing Cname, No_of_orders, Avg_order_amt, where the middle column is the total number of orders by the customer and the last column is the average order amount for that customer.

d. List the orders that were not shipped within 30 days of ordering.

e. List the Order# for orders that were shipped from *all* warehouses that the company has in New York.

Q3. [Use MySQL or a similar software for this problem]

Use the schema provided in Figure 5.5 of your textbook to create database and populate it with sufficient sample data. Provide a description of your create table and insert queries (just a few inserts) you used to create the aforementioned database. If this step is missing then I will not grade rest of the solution. Remember to work with PKs & FKs.

Now specify the following queries in **SQL** and also take print screen of the output to associate with each query:

a. Retrieve the names of all employees who work in the department that has the employee with the highest salary among all employees.

b. Retrieve the names of all employees whose supervisor's supervisor has '888665555' for Ssn.

c. Retrieve the names of employees who make at least $10,000 more than the employee who is paid the least in the company.

Q4.

Orders

Number	order_date	cust_id	salesperson_id	Amount
10	8/2/96	4	2	540
20	1/30/99	4	8	1800
30	7/14/95	9	1	460
40	1/29/98	7	2	2400
50	2/3/98	6	7	600
60	3/2/98	6	7	720
70	5/6/98	9	7	150

Salesperson

ID	Name	Age	Salary
1	Abe	61	140000
2	Bob	34	44000
5	Chris	34	40000
7	Dan	41	52000
8	Ken	57	115000
11	Joe	38	38000

Customer

ID	Name	City	Industry Type
4	Samsonic	pleasant	J
6	Panasung	oaktown	J
7	Samony	jackson	B
9	Orange	Jackson	B

* In orders table cust_id is a foreign key referencing ID of Customer table, and salesperson_id is a foreign key referencing ID of Salesperson table.

* Number, ID, ID are primary keys for Orders, Salesperson, Customer tables respectively.

Provide SQL queries for the following:

a) Find the names of that salesperson (s) that have 2 or more orders.

b) Find the **name** of the customer that has constituted to the highest amount (total amount) for the orders.

Q5.

Consider the following set of requirements for a UNIVERSITY database that is used to keep track of students' transcripts.

(a) The university keeps track of each student's name, student number, social security number, current address and phone, permanent address and phone, birthdate, sex, class (freshman, sophomore, ..., graduate), major department, minor department (if any), and degree program (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city, state, and zip of the student's permanent address, and to the student's last name. Both social security number and student number have unique values for each student.

(b) Each department is described by a name, department code, office number, office phone, and college. Both name and code have unique values for each department.

(c) Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of course number is unique for each course.

(d) Each section has an instructor, semester, year, course, and section number. The section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester.

(e) A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3, 4 for F, D, C, B, A, respectively).

Design an ER schema diagram for this application, you are free to make reasonable assumptions but make sure to clearly state any and all assumptions you make. Label the diagram using (min, max) as you have learned in the lectures.

Q6.

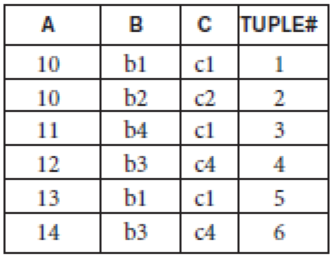
Consider the relation R = {A, B, C, D, E, F, G, H, I, J} and the set of functional dependencies

F = {{A, B}->{C}, {A}->{D, E}, {B}->{F}, {F}->{G, H}, {D}->{I, J}}.

What is the key for R? Decompose R into 2NF and then 3NF relations.

Q7.

Consider the following relation:



A	B	C	TUPLE#
10	b1	c1	1
10	b2	c2	2
11	b4	c1	3
12	b3	c4	4
13	b1	c1	5
14	b3	c4	6

a. Given the previous extension (state), which of the following dependencies may hold in the above relation? If the dependency cannot hold, explain why by specifying the tuples that cause the violation.

i. A -> B, ii. B -> C, iii. C -> B, iv. B -> A, v. C -> A

b. Does the above relation have a potential candidate key? If it does, what is it? If it does not, why not?

Q8.

Earlier you designed an ER diagram for the University database. Include that ER diagram in your solution for this assignment as well. Based on your ER and understanding of relational database, convert the University ER diagram to a database with required number of tables. Each table must be normalized to 3rd Normal Form. Include given (and assumed) functional dependencies (FD+) and closures (FD+) as well. Show the normalization steps from 1NF -> 3NF.

Solution must contain:

a. ER diagram
b. Tables normalized up to 3NF
c. FDs & FD+

Q9.

Suppose a data file for Student (Name, SSN, Address, Phone, Email…). Given that the data file's record size is 400 bytes, disk block size is 1024 bytes, and total number of records are 3000000. For index entry, the field size is 9 bytes, and the pointer size is 8 bytes. Answer the following questions:

a. What will be the average linear search cost using data file?

b. What will be binary search cost using data file?

c. What will be binary search cost using index file (dense index)?

Discuss why using indexes is almost always better.

Q10.

What are two major differences between B and B+ trees? Show step by step insertion in a B+ tree with the insertion sequence as: 7, 3, 12, 9, 8. Then, show how deletion would happen in the same B+ tree with the sequence as: 8, 3.

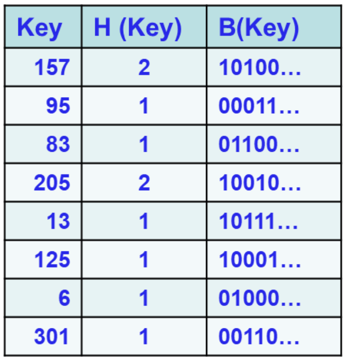
use p=3, pleaf=2. refer to the figure 17.12 on pg 628 of the textbook

Q11.

Using dynamic hashing, demonstrate step by step how the following records would be inserted (show the state at each step). The keys for the records to be inserted are in the order as:

95, 83, 6, 157, 301, 125, 205, 13 – so the record with key 95 is inserted first, and then the record with the key 83, and then so on and so forth. Bucket size = 3.

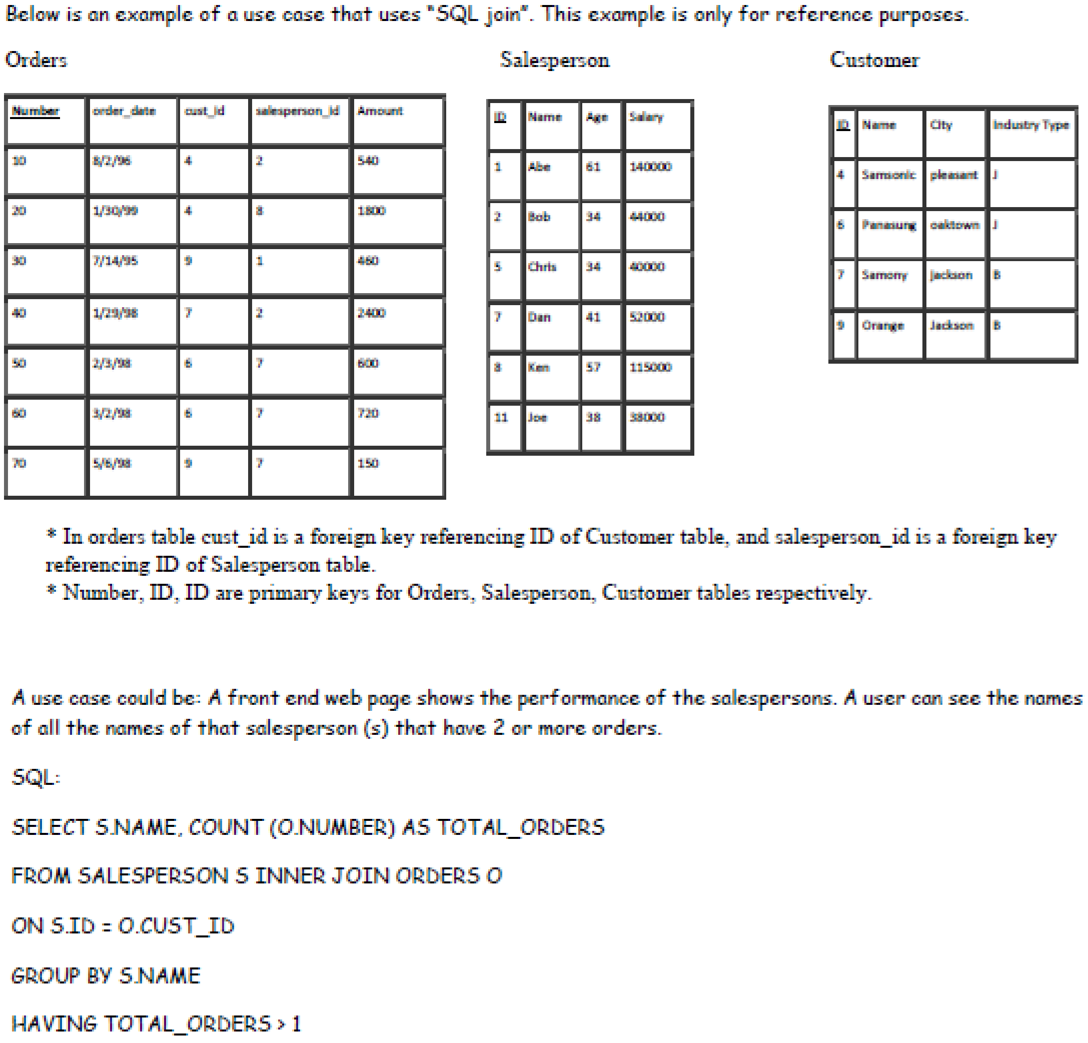
The H & B key table is:



Key	H (Key)	B(Key)
157	2	10100...
95	1	00011...
83	1	01100...
205	2	10010...
13	1	10111...
125	1	10001...
6	1	01000...
301	1	00110...

Q12.

Based on the solutions of Assignment#2 & Assignment#4, design 3 use that would involve running SQL query for the database you have designed. For each use case you design make sure it uses "SQL join". Explain the use cases in brief and then present the SQL query involved in each.



Below is an example of a use case that uses "SQL join". This example is only for reference purposes.

Orders

Number	order_date	cust_id	salesperson_id	Amount
10	8/2/96	4	2	540
20	1/30/99	4	8	1800
30	7/14/95	9	1	460
40	1/29/98	7	2	2400
50	2/3/98	6	7	600
60	3/2/98	6	7	720
70	5/6/98	9	7	150

Salesperson

ID	Name	Age	Salary
1	Abe	61	140000
2	Bob	34	44000
5	Chris	34	40000
7	Dan	41	52000
8	Ken	57	115000
11	Joe	38	38000

Customer

ID	Name	City	Industry Type
4	Samsonic	pleasant	J
6	Panasung	oaktown	J
7	Samony	jackson	B
9	Orange	Jackson	B

* In orders table cust_id is a foreign key referencing ID of Customer table, and salesperson_id is a foreign key referencing ID of Salesperson table.
* Number, ID, ID are primary keys for Orders, Salesperson, Customer tables respectively.

A use case could be: A front end web page shows the performance of the salespersons. A user can see the names of all the names of that salesperson (s) that have 2 or more orders.

SQL:

```
SELECT S.NAME, COUNT (O.NUMBER) AS TOTAL_ORDERS

FROM SALESPERSON S INNER JOIN ORDERS O

ON S.ID = O.CUST_ID

GROUP BY S.NAME

HAVING TOTAL_ORDERS > 1
```

Q13.

Design your MongoDB databases to satisfy the problem statement of Assignment#2. Also, consider the use cases you designed in problem above and the solutions of Assignment#2. Now present your argument and reasoning that your design and schema is well suited for the problem statement and the 3 use cases. Include a sample document from each collection. Now present your Mongo query for each of the 3 use cases. Show your work for maximum credit.