SWE3003 Introduction to Database Systems - Midterm Spring 2023

Student ID	Name

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Total
For Instructor/TA only,										

Academic Honor Pledge

I affirm that I will not at any time be involved with cheating or plagiarism while enrolled as a student at SungKyunKwan University.
I understand that violation of this code will result in penalties as severe as indefinite suspension from the university.

Your signature: _____

1. [20 pts] For each of the following statements, indicate whether it is TRUE or FALSE. You will get 2 points for each correct answer, -2 points for each incorrect answer, and 0 point for each answer left blank or both answers marked.

		T	F
(a)	Each row in a database table can be uniquely identified by a super key. T	☐	☐
(b)	Normalization is the process of organizing data in a database to eliminate redundant data. ..... T	☐	☐
(c)	The special value null is a member of every domain. ..... T	☐	☐
(d)	Given relations R(A, B) and S(A, C), the full outer join $R \text{⟗} S$ is equal to $R \cup (R \cap S) \cup S$. ..... F	☐	☐
(e)	In the WHERE clause of a SQL query, the condition NULL <> NULL is evaluated to be false. ..... T	☐	☐
(f)	When an attribute A in a table T is declared as a foreign key that references a table R, the DBMS checks the foreign-key constraint each time a tuple in T is read. ..... F	☐	☐
(g)	Every weak entity set in an entity-relationship model requires at least one total participation relationship. ..... T	☐	☐
(h)	If a relation R is in BCNF, R is also in 3NF. ..... T. BCNF $\subset$ 3NF	☐	☐
(i)	Normalization to 3NF (Third Normal Form) yields less redundancy than normalization to BCNF (Boyce-Codd Normal Form). ..... F	☐	☐
(j)	The data in a table with a clustered index is stored on disk in the same order as the index. ..... T	☐	☐

2. [10 pts] Consider the following DB schema for an online store:

Customers table:

customer_id	name
alice	Alice Wonderland
bob	Bob Cat
charlie	Charlie Brown
dave	Davey Crockett
emma	Emma Lou
frank	Franken Stein

Orders table:

order_id	customer_id	book_id
1	alice	10
2	bob	10
3	charlie	20
4	dave	30
5	charlie	10
6	bob	20
7	dave	10

Books table:

book_id	title	price
10	Atomic Habits	2500
20	Beyond SQL	1500
30	Consistency Factor	3500
40	Database Design	1500

Show the output relation for each of the following relational algebras.

- (a) $\Pi_{name,title,price}(\sigma_{name='Bob'}(Customers \rtimes (Orders \bowtie Books)))$

  answer:

name	title	price
Bob Cat	Atomic Habits	2500
Bob Cat	Beyond SQL	1500

- (b) $\Pi_{name,title,price}(\sigma_{name='Bob'}((Customers \rtimes Orders) \bowtie Books))$

  answer:

name	title	price
Bob Cat	Atomic Habits	2500
Bob Cat	Beyond SQL	1500

3. [35 pts] Write each of the following queries in SQL for the given tables.

Customers table:

customer_id	name
alice	Alice Wonderland
bob	Bob Cat
charlie	Charlie Brown
dave	Davey Crockett
emma	Emma Lou
frank	Franken Stein

Orders table:

order_id	customer_id	book_id
1	alice	10
2	bob	10
3	charlie	20
4	bob	40
5	dave	30
6	charlie	10
7	bob	30
8	dave	10
9	bob	20

Books table:

book_id	title	price
10	Atomic Habits	2500
20	Beyond SQL	1500
30	Consistency Factor	3500
40	Database Design	1500

(a). Select the name of customers who purchased a book titled "Beyond SQL".

answer (5 pts):

```
SELECT c.name
FROM Customers c
NATURAL JOIN Orders o
NATURAL JOIN Books b
WHERE b.title = 'Beyond SQL';
```

(b). Select the total amount spent by each customer on their orders:

answer (5 pts):

```
SELECT c.customer_id, c.name, SUM(b.price)
FROM customers c
NATURAL JOIN orders o
NATURAL JOIN books b
GROUP BY c.customer_id, c.name;
```

(c) Write a SQL query to list all customers who have placed at least one order.

answer (5 pts):

```
SELECT c.name
FROM Customers c
NATURAL JOIN Orders o
GROUP BY c.customer_id;
```

Customers table:

customer_id	name
alice	Alice Wonderland
bob	Bob Cat
charlie	Charlie Brown
dave	Davey Crockett
emma	Emma Lou
frank	Franken Stein

Orders table:

order_id	customer_id	book_id
1	alice	10
2	bob	10
3	charlie	20
4	bob	40
5	dave	30
6	charlie	10
7	bob	30
8	dave	10
9	bob	20

Books table:

book_id	title	price
10	Atomic Habits	2500
20	Beyond SQL	1500
30	Consistency Factor	3500
40	Database Design	1500

(d) Write a SQL query to find any customer who has bought ALL books that cost more than 2500.

answer (10 pts):

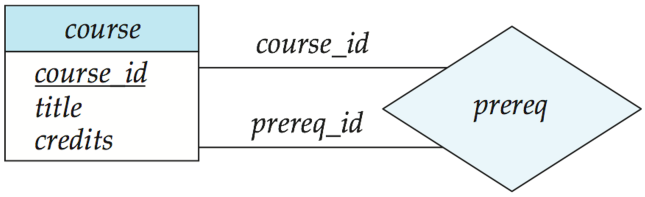
```
SELECT o.customer_id
FROM Orders o
NATRUAL JOIN Books b
WHERE b.price > 2500
GROUP BY o.customer_id
HAVING COUNT(DISTINCT o.book_id) =
            (SELECT COUNT(*)
             FROM Books
             WHERE price > 2500);
```

(e) Write a SQL query to find the customer who has spent the most money on orders.

answer (10 pts):

```
SELECT c.name, SUM(b.price)
FROM Customers c
NATRURAL JOIN Orders o
NATRURAL JOIN Books b
GROUP BY c.customer_id
HAVING SUM(b.price) =
        (SELECT MAX(total_spent)
         FROM (SELECT SUM(b.price) as total_spent
               FROM Customers c
               NATURAL JOIN Orders o
               NATURAL JOIN Books b
               GROUP BY c.customer_id));
```

4. [10 pts] Write SQL DDL statements for the following ER-diagram.
   Note: You may assume any type for all attributes.



answer:

```
CREATE TABLE Course (
  course_id INT PRIMARY KEY,
  title VARCHAR(255),
  credits FLOAT
);

CREATE TABLE Prerequisite (
  course_id INT,
  prereq_id INT,
  PRIMARY KEY (course_id, prereq_id),
  FOREIGN KEY (course_id) REFERENCES Course(course_id),
  FOREIGN KEY (prereq_id) REFERENCES Course(course_id)
);
```

5. [15 pts] Consider the following relations and functional dependencies (FDs) below.

(a) Is the following R in BCNF? If not, decompose it into BCNF.

```
R(A, B, C, D, E)
A,B → C,D,E
C → D
```

answer:

No.

```
R1(A, B, C, E)
R2(C, D)
```

(b) Is the following R in 3NF? If not, decompose it into 3NF.

```
R(A, B, C, D, E)
A → B
B → C
C → D
A → E
```

answer:

No.

```
R1(A, B, E)
R2(B, C)
R3(C, D)
```

(c) Is the following R in BCNF? If not, decompose R into BCNF.

```
R(A, B, C, D, E)
A → BC
B → D
C → E
E → A
```

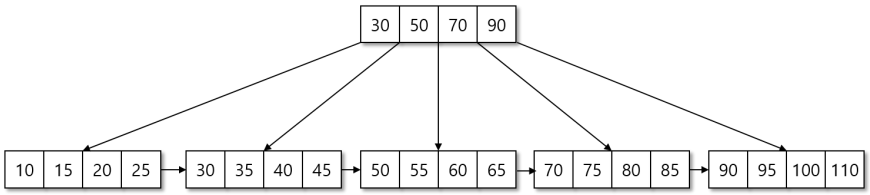
answer:

No. B → D violates BCNF.

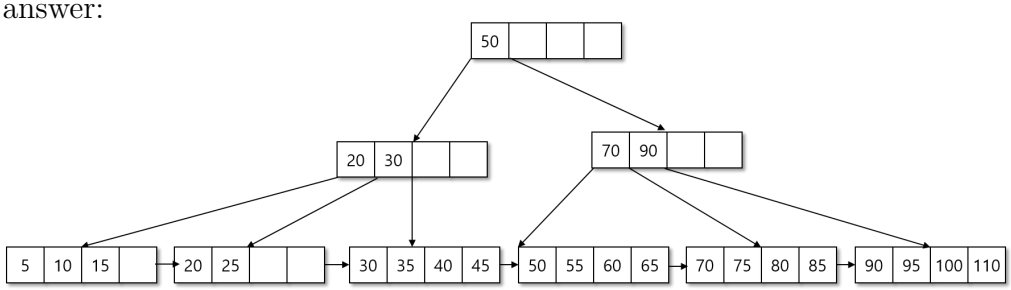
```
R1(A, B, C)
R2(B, D)
R3(C, E)
Uh.oh.. we lost FD: E -> A.
```

6. [10 pts] How B+tree works.

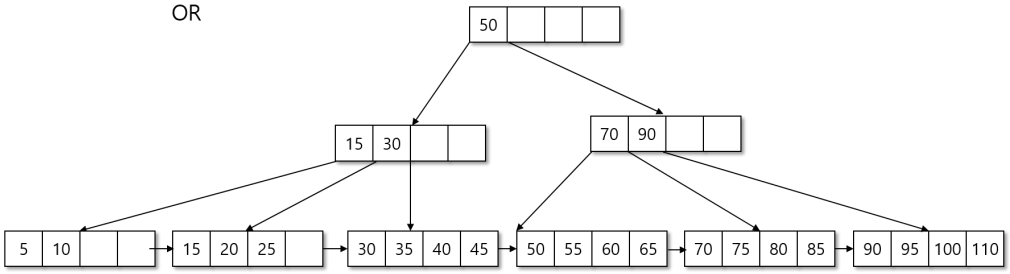
(a) Below is the state of B+tree with degree 5. Draw the state of the B+tree after inserting 5.



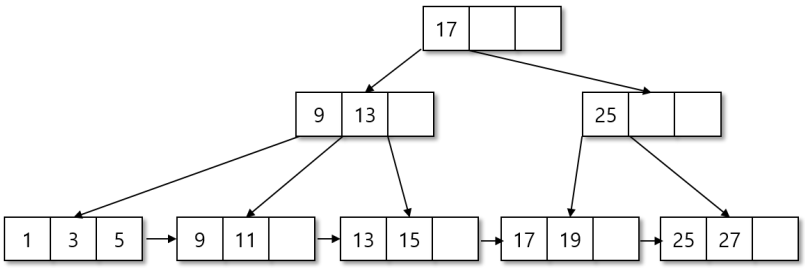
answer:



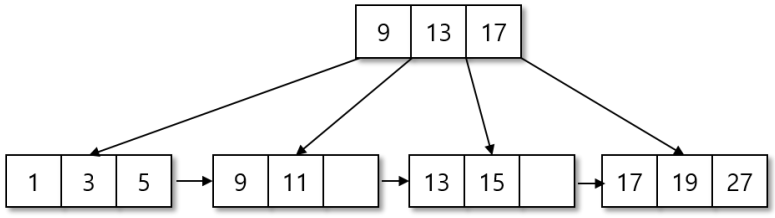
OR



(b) Below is the state of B+tree with degree 4. Draw the state of the B+tree after deleting 25.

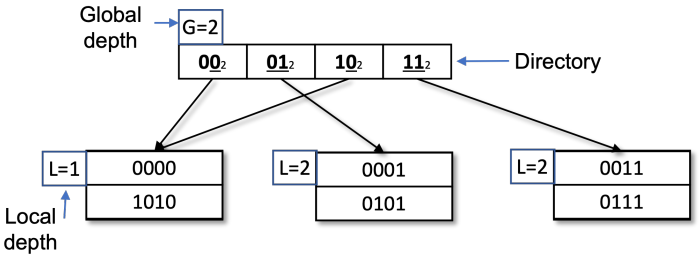


answer:

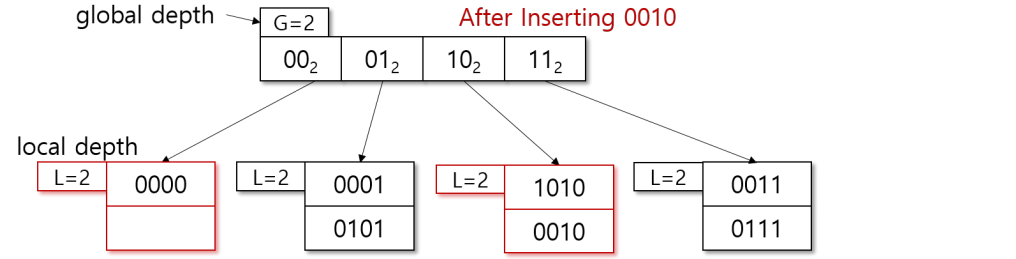


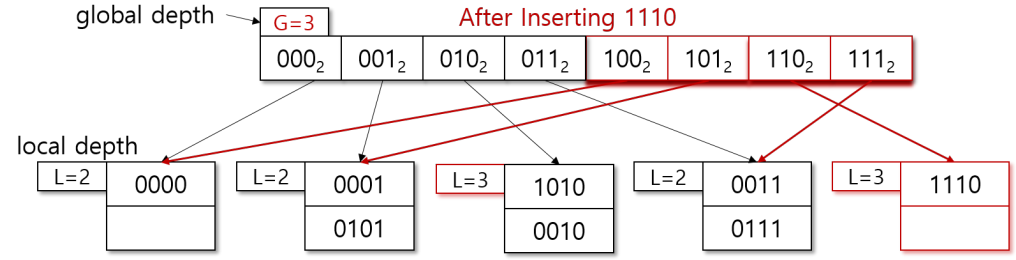
7. [5 pts] Below is the current state of extendible hash table. We use the least significant bits of keys and $\%2^G$ as the hash function. Draw a diagram showing the state of the extendible hash table after inserting the following 2 more data entries.

   2 (0010_2) and 14 (1110_2)

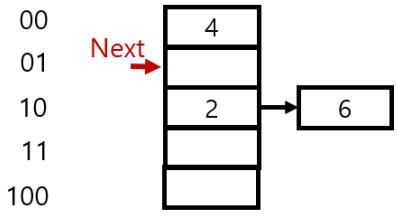


answer:

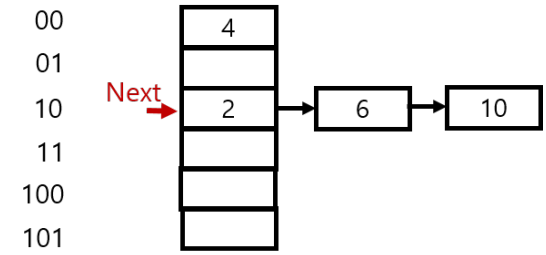


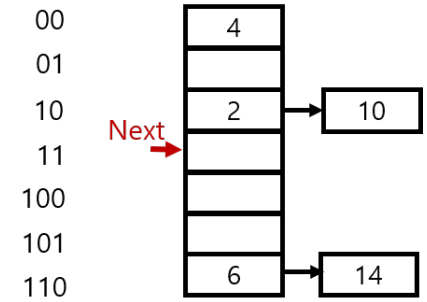


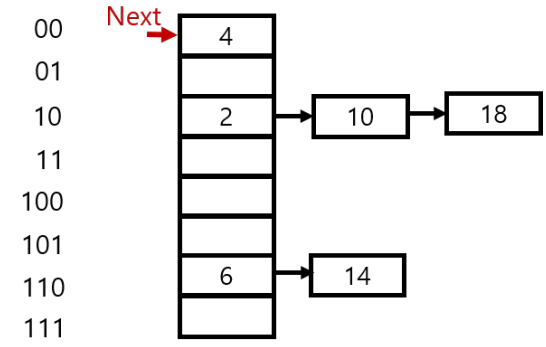
8. [5 pts] Below is the current state of linear hash table, where each bucket can hold only one key. We use the least significant bits of keys and $\%2 \cdot 2^{level}$ as the hash function. Draw a diagram showing the state of the linear hash table after inserting 10, 14, and 18



answer:







9. [10 pts] Consider the following table.

ID	Name	Gender	Age
1	Alice	Female	25
2	Bob	Male	30
3	Carol	Female	35
4	David	Male	40

Table 1: Caption

(a) Create bitmap indexes on the "Gender" and "Age" attributes.

answer:

```
Male: 0101
Female: 1010

Age0_9: 0000
Age10_19: 0000
Age20_29: 1000
Age30_39: 0110
Age40_over: 0001
// You may divide the age range as you like
```

(b) Show how your bitmap indexes can be used to optimize the following query? Explain the steps involved.

```
SELECT * FROM table WHERE Gender = 'Female' AND Age > 30;
```

answer:

You can perform a bitwise AND operation on the Female bitmap index and Age range bitmap index. For example,

```
selected = Female_index & ( Age30_39 | Age40_over )
while(selected){
   if(selected & 1) {
       // select this record
   }
   selected >> 1;
}
```