

1. **(10 points) What is the role of a DBMS, and what are its advantages?**

The role of DBMS is to manage your data within the database. You can say it is a set of programs to access the data. And the advantages of it is convenient and efficient for us to manipulate the data.

2. **(10 points) Please describe the differences between “Logical data independence” and “physical data independence”.**

Logical Data Independence focuses on the ability to change the conceptual scheme without changing external views or API programs. Physical Data Independence focuses on the ability to modify the physical schema without changing the logical schema

3. **(10 points) Please describe the processing flowchart of a SQL query in a database system**

key words: Query Processing

1. **Parsing and translation:** 接收到查詢後，會檢查語法是否正確
2. **Optimization:** Optimizer 會找到並執行查詢的最有效方法
3. **Evaluation:** 會有一個查詢計劃，查詢引擎開始執行操作，訪問存儲系統以獲取和處理數據。最後，處理後的數據被返回給用戶或應用程序

4. **(15 points) Please describe the differences among shared memory, shared disk, shared nothing, and hierarchical architectures in parallel database architectures**

1. shared memory: processors share a common memory
2. shared disk: processors share a common set of disks, also called clusters.
3. shared nothing: processors share neither a common memory nor common disk
4. hierarchical: hybrid of the above three architectures

5. **(15 points) Please use an example to describe the differences in meaning among the terms superkey, candidate key, primary key, and foreign key. And list the reasons why null values should be introduced into the database system.**

1. superkey: the attribute set that can identify unique row
2. candidate key: the smallest superkey. If you remove any attribute from candidate keys, it is no longer has the unique identified ability of each data
3. primary key: choose from candidate key, used to identify each row from the relations
4. foreign key: make sure the reference integrity is between two relations, a value in a relation can match primary key in another relation
5. null values: signifies that the value is unknown or does not exist

6. (20 points) Given two relations instructor and teaches, please write down the SQL according to the following query

<i>instructor</i>				<i>teaches</i>				
<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>	<i>ID</i>	<i>course_id</i>	<i>sec_id</i>	<i>semester</i>	<i>year</i>
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2009
12121	Wu	Finance	90000	10101	CS-315	1	Spring	2010
15151	Mozart	Music	40000	10101	CS-347	1	Fall	2009
22222	Einstein	Physics	95000	12121	FIN-201	1	Spring	2010
32343	El Said	History	60000	15151	MU-199	1	Spring	2010
33456	Gold	Physics	87000	22222	PHY-101	1	Fall	2009
45565	Katz	Comp. Sci.	75000	32343	HIS-351	1	Spring	2010
				45565	CS-101	1	Spring	2010
				45565	CS-319	1	Spring	2010
				76766	BIO-101	1	Summer	2009
				76766	BIO-301	1	Summer	2010

- a. In relation instructor, find the names and average salaries of all departments whose average salary is greater than 38000.

dept_name	avg_salary
Comp. Sci.	70000
Finance	90000
Physics	91000
History	60000
Music	40000

```

SELECT dept_name, AVG(salary) AS avg_salary
FROM instructor
GROUP BY dept_name
HAVING AVG(salary) > 38000;

```

- b. Find the instructor whose name containing the word "E".

Einstein

El Said

```
SELECT * FROM instructor WHERE name LIKE '%E%';
```

- c. List the names of instructors along with the course_id of courses that they teach.

Srinivasan CS-101

Wu FIN-201

Mozart MU-199

Einstein PHY-101

El Said HIS-351

Gold No course id

Katz CS-319

```
SELECT instructor.name, teaches.course_id
FROM instructor
JOIN teaches ON instructor.ID = teaches.ID;
```

- d. Find the course_id that is taught in Fall 2009 and Spring 2010 using Intersect

CS-101

```
SELECT course_id
```

```
FROM teaches
```

```
WHERE semester = 'Fall' AND year = 2009
```

```
INTERSECT
```

```
SELECT course_id
```

```
FROM teaches
```

WHERE semester = 'Spring' AND year = 2010;

7. (20 points) Consider the following Table Definition

```
create table student (  
    ID      char(10),  
    name    varchar(30) not null,  
    dept_name varchar(20),  
    age     int(3),  
    primary key (ID),  
    foreign key (dept_name) references department,  
    check (age >= 0));
```

a. What is the purpose to define: not null ?

Make sure the specific column cannot have null value

b. What is the purpose to define: primary key (ID) ?

Used to identify the unique row

c. What is the purpose to define: foreign key (dept_name) references department?

Used to build up the relationship between two tables.

d. What is the purpose to define: check (age >= 0)?

Impose constraint on data can make sure the value is stored correctly in the column.

If you want to input a value for “age” attribute, age is always positive in the real world.

As a result, check(age>=0) could prevent some unreasonable value