Database Management, 2020 Midterm

Q1. (6%) Explain the **entity integrity constraint, foreign key** and the **referential integrity constraint**.

Q2.
(a) (4%) Draw diagrams to explain the three-tier client server architecture.
(b) (3%) When a NULL is involved in a comparison operation, the result is considered to be UNKNOWN. Show the results of the following logical expression respectively: (i) TRUE AND（FALSE OR UNKNOWN）; (ii) (TRUE OR UNKNOWN) AND UNKNOWN; (iv) (TRUE AND UNKNOWN) OR (UNKNOWN OR UNKNOWN).

Q3. (4%) Explain inherent (implicit) constraints and schema-based (explicit) constraints.

Q4. Aggregate functions can be applied to a particular column (attribute), that is, a collection of values.
(a) (4%) Explain how the NULL values are handled when the SUM function is applied to the attribute Hours in the WORKS_ON table. If the collection becomes empty because all values are NULL, what will the SUM function return?
(b) (2%) If the collection becomes empty because all values are NULL, what will the COUNT function return?
(c) (2%) Explain how will the COUNT(*) handle the tuple (of the query result) that contain NULL values of some attributes in the tuple?

Q5. Below are three tables for “2020 Best Singer Battle!!” in NCTU:

STUDENT

Student_number	Name	Sex	Major
5	Tony	male	MIS
16	Kelly	female	FL
49	Jay	male	EE

SONG

Song_id	Language	Type	Producer
2	Taiwanese	Electropop	SeedMusic
11	Chinese	Pop	BinMusic
17	English	Country	SouthernMusic
34	English	Blues	SonyMusic
82	Chinese	R&B	EnjoyMusic

COMPETITION

Student_number	Song_id	Score
49	82	B
16	2	A
49	34	C
5	17	B
5	11	B

Write SQL update statement to do the following on the database schema shown in above Figure.

a. (4%) Update all the Competition Scores of the songs, in which the Language is ’English’ and is sang by male student to ‘A’.

b. (3%) Insert a new song, <77,’English’,’Pop’,’BinMusic’>

c. (3%) Delete records from the Competition table, in which the song is sang by Kelly.

d. (5%) Write a SQL query to list the Song_id of each song that got at least two scores of ‘B’ and was sang by at least two MIS students in the competitions.

Q6. Below is a subset of relations from COMPANY schema. The keys have been underlined.
EMPLOYEE (FNAME, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO)
DEPARTMENT (DNAME, DNUMBER, MGRSSN, MGRSTARTDATE)
PROJECT (PNAME, PNUMBER, PLOCATION, DNUM)
WORKS_ON (ESSN, PNO, HOURS)
DEPENDENT (ESSN, DEPENDENT_NAME, SEX, BDATE, RELATIONSHIP)

Express the following Queries in SQL statements.

(1) (6%) Query 1: For each employee with more than three direct supervisees, list the name and the total number of direct supervisees of the employee.

(2) (6%) Query 2: For each employee who is not a manager and has more than two dependents, list the name of the employee, the name of his/her supervisor and the number of projects that the employee works on.

(3) (6%) Query 3: For each department that controls more than five projects, retrieve the name of the department, the name of the department manager, and the number of Male employees who work for the department.

(4) (6%) Query 4: For each department manager who has no dependents and whose salary is less than the salaries of some employees in his/her department, list the name of the manager and the name of the department.

(5) (6%) Query 5: For each project whose number of working employees is greater than the number of working employees of every project controlled by department number 2, list the name of the project and the name of its controlling department.

(6) (6%) Query 6: Retrieve the name and location of each project that is controlled by “R&D” department and has at least three employees of “R&D” department working on the project.

(7) (6%) Query 7: Retrieve the name of each employee who works on all the projects located in Hsinchu and controlled by MIS department.

(8) (8%) Query 8: For each employee who works for the Research department and has more than two supervisees (direct and indirect supervisees), list the name of the employee and the names of all his/her supervisors (direct and indirect supervisors) who are not the department managers.

Q7. Given the following PHP program:

```
. . . . . .
print <<<_HTML_
<FORM method = "post" action ="$_SERVER['PHP_SELF']">
Enter department name: <INPUT type ="text" name="dept_name"> <BR/>
Enter employee salary threshold: <INPUT type ="text" name="emp_sal"> <BR/>
<INPUT type="submit" value="submit your input">
</FORM>
_HTM_;
. . . . .
$q = $d->query("SELECT FNAME, LNAME, SALARY FROM EMPLOYEE, DEPARTMENT
WHERE DNAME =  (1)  AND SALARY  >  (2)   AND  (3)  ");
print "employees work for department  (4) , and employee salary is greater than  (5) : \n"
while ($r =  ______(6)_____ ) {
        print "employee first name:  (7) , last name:  (8) , salary:  (9) \n"
}
```

(a) (6%) Fill in the blanks of (1), (2) and (3).

(b) (2%) Fill in the blanks of (4) and (5).

(c) (4%) Fill in the blanks of (6), (7), (8) and (9).

Q8. Assume that the array variable $supervising is associative, and each element in $supervising associates an employee name (key) with the supervisor of the employee (value). Suppose that the associations of employees and supervisors are as follows. 'Kevin' is the supervisor of 'John'; 'Tim' is the supervisor of 'Darrel'; 'Jack' is the supervisor of 'Mary'.

(a) (3%) Write a PHP statement to assign the employee-supervisor associations to the array variable $supervising.

(b) (5%) Write a PHP program looping through all the elements in $supervising using the **foreach** construct, and printing the employee name and supervisor name of each element on a separate line.

Q1.

Entity Integrity(一致性) constraint：

no primary key value can be NULL. This is because the primary key value is used to identify individual tuples in a relation. Having NULL values for the primary key implies that we cannot identify some tuples. For example, if two or more tuples had NULL for their primary keys, we may not be able to distinguish them if we try to reference them from other relations.

Primary key 代表 entity，所以 primary key 不能為 NULL

Primary key 是代表資料庫每一筆 tuple 的 idntity，所以 primary key 不能有 NULL 因為 domain constraints，key 是 unique 的

The primary key attributes PK of each relation schema R in S cannot have null values in any tuple of r(R).

This is because primary key values are used to identify the individual tuples.

t[PK] ≠ null for any tuple t in r(R)

If PK has several attributes, null is not allowed in any of these attributes

Note: Other attributes of R may be constrained to disallow null values, even though they are not members of the primary key.

foreign key：

reference the primary key attributes PK of the another referenced relation R2.
If a relation schema includes the primary key of another relation schema, that attribute is called the foreign key

Reference 其他 table 的 primary key 的 key

EX：

EMPLOYEE

SSN	SUPERSSN	DNO
e1	e6	2
e3	e4	2
e4	e5	3

WORK_ON

ESSN	PNO	HOURS
e1	p1	5
e3	p1	8
e4	p3	7
e5	p4	6

DEPT

DNumber	Dname	MGRSSN
1	Develop	e21
2	Design	e21
3	AI	e39

WORK_ON's ESSN is a foreign key since it's reference to **EMPLOYEE**'s primary key (SSN)
EMPLOYEE's DNO is a foreign key because it is reference to **DEPT**'s primary key (Dnumber)
DEPT's MGRSSN is a foreign key because it is reference to **EMPLOYEE**'s primary key (SSN)

referential integrity constraint：

foreign key either reference 存在的值 or 是 NULL

specified between two relations and is used to maintain the consistency among tuples in the two relations. Informally, the referential integrity constraint states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation

A constraint involving two relations

The previous constraints involve a single relation.

Used to specify a relationship among tuples in two relations:

The referencing relation and the referenced relation.

Tuples in the referencing relation R1 have attributes FK (called foreign key attributes) that reference the primary key attributes PK of the referenced relation R2.

A tuple t1 in R1 is said to reference a tuple t2 in R2 if t1[FK] = t2[PK].

A referential integrity constraint can be displayed in a relational database schema as a directed arc from R1.FK to R2.

Statement of the constraint

The value in the foreign key column (or columns) FK of the the referencing relation R1 can be either:

(1) a value of an existing primary key value of a corresponding primary key PK in the referenced relation R2, or

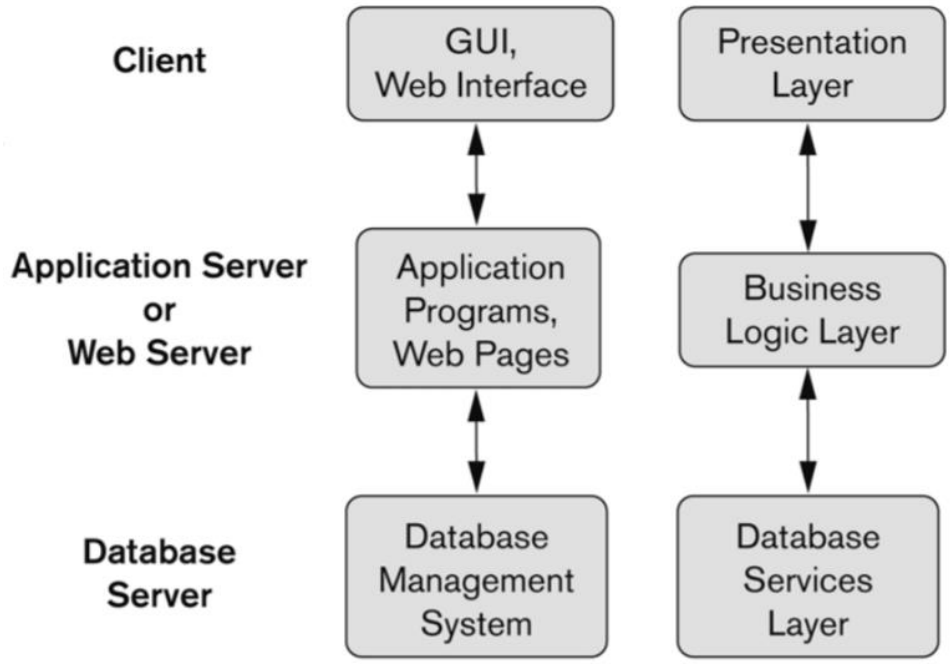
(2) a null.

In case (2), the FK in R1 should not be a part of its own primary key

Q2

(a)

Common for Web applications



Clients

Provide user intervace. Let users can manupulate on it

Receiving user's information and send to Intermediate Layer or presenting information from Intermediate Layer to users

Provide appropriate interfaces through a client software module to access and utilize the various server resources.

Clients may be diskless machines or PCs or Workstations with disks with only the client software installed.

Connected to the servers via some form of a network.

(LAN: local area network, wireless network, etc.)

Represents Web browser, a Java or other application, Applet, WAP phone etc. The client tier makes requests to the Web server who will be serving the request by either returning static content if it is present in the Web server or forwards the request to either Servlet or JSP in the application server for either static or dynamic content.

Intermediate Layer called Application Server or Web Server:

Provide appplication or logic operation

Stores the web connectivity software and the business logic part of the application used to access the corresponding data from the database server

Acts like a conduit for sending partially processed data between the database server and the client.

This layer provides the business services. This tier contains the business logic and the business data. All the business logic like validation of data, calculations, data insertion etc. Are centralized into this tier as opposed to 2-tier systems where the business logic is scattered between the front end and the backend. The benefit of having a centralized business tier is that same business logic can support different types of clients like browser, WAP (Wireless

Application Protocol) client, other standalone applications written in Java, C++, C# etc. This acts as an interface between Client layer and Data Access Layer. This layer is also called the intermediary layer helps to make communication faster between client and data layer

Server

Provide access to database

Provides database query and transaction services to the clients

Relational DBMS servers are often called SQL servers, query servers, or transaction servers

Applications running on clients utilize an Application Program Interface (API) to access server databases via standard interface such as:

ODBC: Open Database Connectivity standard

JDBC: for Java programming access

This layer is the external resource such as a database, ERP system, Mainframe system etc. responsible for storing the data. This tier is also known as Data Tier. Data Access Layer contains methods to connect with database or other data source and to perform insert, update, delete, get data from data source based on our input data

Three-tier Architecture Can Enhance Security:

Database server only accessible via middle tier

Clients cannot directly access database server

Clients contain user interfaces and Web browsers

The client is typically a PC or a mobile device connected to the Web

High performance, lightweight persistent objects.

Scalability – Each tier can scale horizontally.

Performance – Because the Presentation tier can cache requests, network utilization is minimized, and the load is reduced on the Application and Data tiers.

Better Re-usability.

Improve Data Integrity.

Improved Security – Client is not direct access to database.

Forced separation of user interface logic and business logic.

Business logic sits on small number of centralized machines (may be just one).
Easy to maintain, to manage, to scale, loosely coupled etc.

(b)

(i) TRUE AND（FALSE OR UNKNOWN）→ TRUE AND UNKNOWN → UNKNOWN

(ii) (TRUE OR UNKNOWN) AND UNKNOWN → TRUE AND UNKNOWN → UNKNOWN

(iv) (TRUE AND UNKNOWN) OR (UNKNOWN OR UNKNOWN)
→ UNKNOWN OR UNKNOWN → UNKNOWN

Q4

(a)

通常不算 NULL 值

WORKS_ON

Hour
8
10
8
12
NULL
10

Sum: 48

If the collection becomes empty because all values are NULL, SUM will return NULL

(b)

全部都是 empty ➔ Count return 0
如果是用 Agggregate function 且全部都是 NULL ➔ NULL

(c)

COUNT(*) ：

will return the total of all records returned in the result set regardless of NULL values.
counts the number of rows.
Count the number of tuple not of attribute ➔ the tuple would be counted in

WORKS_ON

Hour
8
10
8
12
NULL
10

COUNT (*)： 6

Q5

a

UPDATE COMPETITION
SET Score = ‘A’

```
WHERE Song_id IN (SELECT Song_id
                  FROM SONG
                  WHERE Type = 'Pop')
      AND Student_number IN (SELECT Student_number
                             FROM STUDENT
                             WHERE Name = 'Tony');
```

b

```
INSERT INTO SONG
VALUES (77,'English','Pop','BinMusic');
```

c

```
DELETE FROM SONG
WHERE Producer = 'BinMusic' AND Type = 'R&B';
```

d

```
SELECT STUDENT.Name
FROM STUDENT
WHERE STUDENT.Sex = 'male' AND (SELECT COUNT (*)
                                FROM SONG, COMPETITION
                                WHERE SONG.Producer = 'SonyMusic'
                                AND COMPETITION.Song_id = SONG.Song_id
                                 AND COMPETITION.Score = 'B'
                                 AND
                                SONG.Student_number = COMPETITION.Student_number
                                 ) >= 2;
```