DB Management Midterm HW, 110705017 何翊華 資財

Q1

(a)

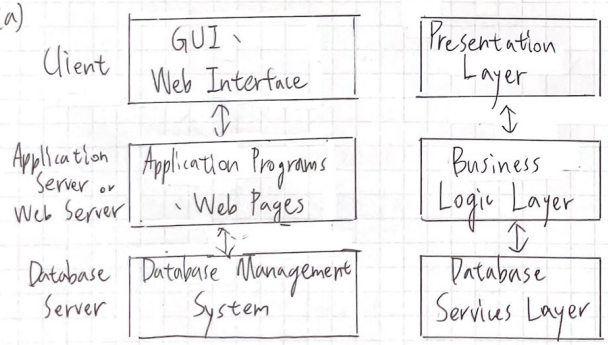
Entity integrity constraint: Primary key must be unique and not "NULL".
Foreign key: The key that reference the primary key.
Referential integrity constraint: The value in foreign key column of original relation can be ① a value of an existing primary key value of corresponding primary key in referenced relation, or ② "NULL".

(b)

If we put all the attributes of Department table in Employee table, we are wasting space. Because the values of attribute DNO are highly repeated, thus the other attributes in Department table are also the same, which is not necessary to list them all.

Q2

(a)



(b)

(i) TRUE

(ii) UNKNOWN

(iii) TRUE

Q3

(a)

Semantic constraints: Beyond the expressive power of model and must specified and enforced by application programs.

Schema-based constraints: Express in schema by using facilities provided by model.

(b) ex.

CAR(State, Reg#, SerialNo, Make, Model, Year)

Minimal superkey = {SerialNO}

Maximal superkey = {State, Reg#, SerialNo, Make, Model, Year}

Q4

(a)

AVG function return nothing , COUNT function return 0.

(b)

COUNT(salary) accumulate each record in table that's not "NULL".

COUNT(distinct salary) calculate the number of different salary.

Q5

(a)

```
UPDATE COMPETITION
SET SCORE = 'A'
WHERE Song_id IN(SELECT Song_id
                 FROM SONG
                 WHERE PRODUCER = "EnjoyMusic")
```

(b)

```
DELETE FROM COMPETITION
WHERE Song_id INCSELECT Song_id
                FROM SONG
                WHERE Type = "R&B")
AND Student_number IN (SELECT Student_number
                       FROM STUDENT
                       WHERE Major = "MIS")
```

(c)

```
SELECT S.NAME
FROM STUDENT AS S, COMPETITION AS C, SONG AS G
WHERE S.Sex = 'male'
 AND S.Student_number = C.Student_number
 AND COUNT(*)(FROM C WHERE C.Score = 'B')>=2
 AND C.Song_id = G.Song_id
 AND G.Language = 'English'
```

Q6

(1)

```
SELECT E.FNAME, E.LNAME, COUNT(*)
FROM DEPARTMENT AS D, EMPLOYEE AS E
WHERE COUNT(*)(FROM E WHERE E.DNO = D.DNUMBER)>3
```

(2)

```
SELECT P.PNAME, E.FNAME, E.LNAME
FROM PROJECT AS P, EMPLOYEE AS E, DEPARTMENT AS D
WHERE D.DNAME = 'MIS'
 AND P.DNUM = D.DNUMBER
 AND E.DNO = D.DNUMBER
```

(3)

```
SELECT E.FNAME, E.LNAME, COUNT(*)
FROM EMPLOYEE AS E, DEPENDENT AS D, WORKS_ON AS W
WHERE E.SSN= D.ESSN IS NULL
AND (SELECT DISTINCT W.PNO
      FROM E, D, W WHERE E.SSN= W.ESSN)
HAVING (OUNT(*) >2
```

(4)

```
SELECT D.DNAME, E.FNAME, E.LNAME, COUNT(DISTINCT
P.PNUMBER) FROM DEPARTMENT AS D
INNER JOIN EMPLOYEE AS E ON D.MGRSSN = E.SSN
INNER JOIN PROJECT AS P ON D.DNUMBER = P.DNUM
WHERE D.DNUMBER IN (SELECT DNO FROM EMPLOYEE
                    GROUP BY DNO
                    HAVING COUNT(*) > 5 )
GROUP BY D.DNUMBER
```

(5)

```
SELECT M.FNAME, M.LNAME, S.FNAME, S.LNAME
FROM EMPLOYEE AS M
INNER JOIN EMPLOYEE AS S ON M.SUPERSSN = S.SSN
WHERE M.SSN IN (SELECT MGRSSN
                FROM DEPARTMENT
                WHERE DNUMBER NOT IN (SELECT
DNUM FROM PROJECT) AND M.SSN NOT IN (SELECT ESSN
FROM WORKS_ON)
```

(6)

```
SELECT P.PNAME
FROM PROJECT AS P
WHERE (SELECT COUNT(*) FROM WORKS_ON AS W
WHERE W.PNO = P.PNUMBER) > (SELECT MAX(MISCOUNT)
FROM (SELECT COUNT(*) AS MISCOUNT
      FROM WORK_ON AS W2, PROJECT AS P2
      WHERE W2.PNO = P2.PNUMBER
      AND P2.DNUM IN (SELECT DNUMBER FROM
      DEPARTMENT WHERE DNAME = "MIS" GROUP BY P2.DNUM
```

(7)

```
SELECT DISTINCT E.FNAME, E.LNAME
FROM EMPLOYEE AS E, PROJECT AS P
WHERE NOT EXISTS((SELECT(*)
                  FROM WORKS_ON AS W
                  JOIN EMPLOYEE AS E2
                  ON W.ESSN = E2.SSN
                  WHERE W.PNO = P.PNUMBER
                  AND E2.FNAME = 'John'
                  AND E2.LNAME = 'Smith')
        EXCEPT(SELECT(*)
               FROM WORKS_ON AS W2
               WHERE W2.PNO = PNUMBER
               AND W2.ESSN=E.ESSN))
```

(8)

```
WITH RECURSIVE SuperEmployees (SupSSN, EmpSSN) AS
(SELECT SupSSN, EmpSSN
 FROM SupEmp
 UNION ALL
 SELECT SE.SupSSN, E.SSN
 FROM SuperEmployees AS SE
 JOIN SupEmp AS S ON SE.EmpSSN = S.SupSSN
 JOIN Employee AS E ON S.EmpSSN = E.SSN)
 SELECT E.FNAME, E.LNAME, SE2.FNAME, SE2.LNAME
 FROM EMPLOYEE AS E
 JOIN SuperEmployees AS SE1 ON E.SSN = SE1.EmpSSN
 JOIN SuperEmployees AS SE2 ON SE1.EmpSSN = SE2.SupSSN
```

```
WHERE E.SEX = 'Male' AND SE1.SupSSN =
    (SELECT SupSSN
     FROM   SupEmp
     WHERE EmpSSN = E.SSN
     GROUP BY SupSSN
     HAVING (COUNT(*) > 2)
```

Q7

(1) array ($_POST['dept_name'], $_POST['proj_loc'])

(2) $_POST['dept_name']

(3) '$_POST['proj_loc']'

(4) $allresult as $r

(5) $r[0]

(6) $r[1]

Q8

(a) $advising = array ('Kevin' => 'John', 'Tim' => 'Parrel', 'Mary' => 'Jack');

(b) foreach ($advising as $student => $advisor)
{ echo $student . 'is advised by' . $advisor . '
';}