

## Question 1)

در روش file system، داده‌ها در فایل‌های مجزا نگهداری می‌شوند و سیستم‌های جداگانه‌ای به نام سیستم پردازش فایل برای استفاده از فایل‌های داده‌ای طراحی می‌شوند. در این سیستم‌ها هر برنامه‌ای کاربردی تعامل با فایل داده‌ها را در یک نقطه می‌تواند مدیریت کند. اشکالات file system به صورت زیر آمده:

① افزونگی و ناسازگاری داده به دلیل چندین فرمت فایل و تعداد اطلاعات در فایل‌های مختلف

② مشکل در دسترسی داده و نیاز به نوشتن برنامه‌ی جدیدی برای انجام هر کار

③ قیدهای جامعیت به جای آنکه صریحاً بیان شوند، در کد برنامه از نظر پنهان می‌شود. اضافه کردن قیدهای جدید یا تغییر قیدهای موجود به سختی صورت می‌گیرد.

④ ایجاد ناسازگاری به دلیل وجود چندین لایه از قلمه‌های داده

⑤ مشکلات امنیتی به دلیل دسترسی محدود و بدون کنترل متوسط بین کاربران

به دلیل مشکلات بالا، مجموعه‌ی داده‌های برنامه‌های سازمان‌ها از سیستم پایگاه داده (database system) استفاده کنیم. در این روش، داده‌ها به صورت مجتمع در پایگاه داده ذخیره می‌شود. ولی هر کاربر دید خاص خود را نسبت به داده‌ها دارد. کاربران مختلف نتوانند به صورت مشترک، پایگاه داده کار کنند. به دلیل تجمع داده، افزونگی به حداقل ممکن کاهش می‌یابد. طراحی روشی به نام سیستم مدیریت پایگاه داده (DBMS) به عنوان واسطه‌ی بین برنامه‌های کاربردی و پایگاه داده نقش ایفا می‌کند، در نتیجه امنیت داده‌ها در این روش بیشتر است. مزایای استفاده از پایگاه داده به صورت زیر است:

① امکان دسترسی سریع به اطلاعات + کاهش افزونگی + جمعیت و وحدت، کنترل و ذخیره سازی داده‌ها

② اطمینان از صحت داده‌ها و اتصال همزمان و نیز قابلیت اشتراک گذاری داده‌ها توسط چندین کاربر

③ پشتیبانی از قابلیت تراکشن

④ استاندارد سازی اطلاعات و نیز صحت بیشتر داده‌ها و استقلال از برنامه‌های کاربردی

⑤ مقیاس پذیری، منعطف و قابل توسعه و همچنین امکان ایجاد برنامه‌های کاربردی جدید

⑥ تجمیع پذیری و استاندارد سازی + پرهیز از ناسازگاری

⑦ پردازش و بررسی داده‌ها

⑧ امنیت بالا

پس، به خاطر مزیت‌های بالا در پایگاه داده، نقطه‌ی تقاطع از file system استفاده کنیم. علاوه بر موارد بالا، در روش file system، ناسازگاری

زیاد بین داده‌ها وجود دارد و معمولاً احتمال وجود افزونگی داده در سیستم فایل می‌رود (به دلیل امکان وجود داده‌های تکراری). پس به طور کلی، امنیت سیستم کارساده‌ای دارد اما دارای معایبی همچون افزونگی داده‌ها، ناسازگاری داده‌ها و امنیت کمتر است.

تفاوت بین سیستم پایگاه داده و سیستم مدیریت فایل در این است که در پایگاه داده، داده‌ها را به واسطه یک ساختار روی دیسک ذخیره می‌کنیم و فایل در یک سیستم مدیریت فایل، داده بدون استفاده از ساختار روی دیسک ذخیره می‌شوند. پایگاه داده اطلاعات را به اشتراک می‌گذارد و از سیستم مدیریت فایل، انعطاف پذیرتر است.

## Question 2)

زبان‌های تعریف داده‌ها (DDL) و زبان‌های دستکاری داده‌ها (DML) هر دو زیرمجموعه‌های SQL اند و تفاوت کلی بین آن‌ها این است که DDL برای تغییر ساختار پایگاه داده استفاده می‌شود و تنها admin مجاز به استفاده از آن است. اما DML برای مدیریت داده‌ها در پایگاه داده، Query کردن توسط userهای عادی و نیز آنالیز داده‌ها (data analysis) و گاهی Programmerهای اپلیکیشن‌ها استفاده می‌شود. در DDL، در ساختار یارین پایگاه داده می‌توان تغییراتی ایجاد کرد و در DML، می‌توان تغییرات را در اصل ساختار پایگاه داده ایجاد نمود.

### Question 3)

(a) غلط

عمر انتخاب قابلیت جای داری و هر دو کوثری  $SELECT a, b$  و  $SELECT b, a$  به همان هر دو  $a, b$  را بر می گرداند.

$$\sigma_{A \wedge B}(r) = \sigma_{B \wedge A}(r)$$

$$\sigma_A(\sigma_B(r)) = \sigma_B(\sigma_A(r))$$

(b) درست

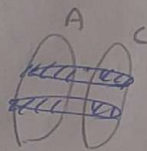
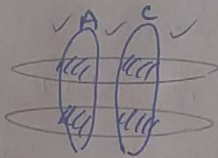
درست است زیرا در عمل  $Project$ ، سطرها را که و یک سری ستون را انتخاب می کنیم و آن چند ستون را انتخاب می کنیم و در عین حال چندین سطر نگه داری وجود داشت، معمولاً آن تعداد را حذف می شوند. پس تایل ها که این نقطه یک بار در فرم خاص می شوند اگر غیر از این بود، یعنی داریم مجموعه تایل ها را تغییر می دهیم که مخالف با مفهوم "key" ها را پیاپی داده است.

(c) غلط

(مثال من)

$$\pi(\sigma_{A=B})_{A|C} \Rightarrow$$

$$\sigma_{A=B}(\pi_{A|C}) \Rightarrow$$



ابتدا تمام سطرها را که در آن ها  $A, B$  برابرند، انتخاب می شوند. پس از میان داده های حاصل شده، فقط ستون های  $A, C$  را انتخاب می کنیم.

در اینجا، ابتدا تایل ستون های  $A, C$  انتخاب می شوند. پس از میان داده های حاصل شده، فقط سطرها را که در آن ها  $A, B$  برابرند را انتخاب می کنیم. همان طوری که در

A	B	C
CE	CE	1
CE	CE	2
Phy	CS	3

$$\Rightarrow \pi(\sigma_{A=B})_{A|C} = \begin{array}{c|c} A & C \\ \hline CE & 1 \\ CE & 2 \end{array}$$

$$\Rightarrow \sigma_{A=B}(\pi_{A|C}) = \begin{array}{c|c} A & C \\ \hline CE & 1 \\ CE & 2 \end{array}$$

پس تفاوتی نیست که انتخاب و پرتو مطابقاً جای جای می بینیم.

به بیان دیگر،  $select$  خاصیت جای جایی دارد ولی  $Project$  نه. پس عمر  $Project$  می تواند در  $select$  زده شود ولی برعکس نه.

(d) درست

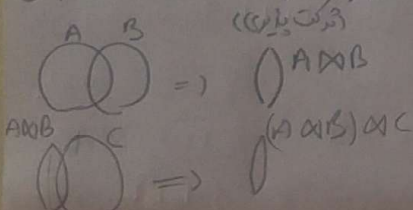
$$\sigma(A \cup B) = (\sigma A) \cup (\sigma B) \Rightarrow$$

(e) درست

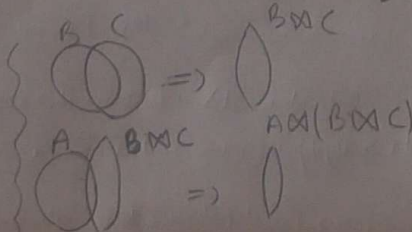
$$\sigma(A \cap B) = (\sigma A) \cap (\sigma B) \Rightarrow$$

(f) درست

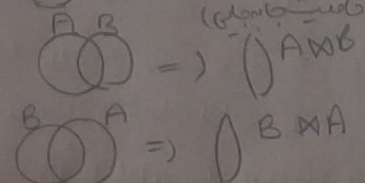
① Associative Property:



natural join و کوثری را دارد. می توان با هم داری و نیز آن را نشان داد:



② Commutative Property:





# Question 4)

(a)  $\Pi$  (Books)  
title, year

SELECT title, year  
FROM Books

3

(b)  $\sigma$  (STUDENTS)  
Field = 'computer Engineering'

SELECT \*  
FROM STUDENTS  
WHERE Field = 'Computer Engineering'

(c)  $\sigma$  (BOOKS)  
(Year < 1398)  $\wedge$  (Publisher = 'Amirkabir')

SELECT \*  
FROM Books  
WHERE (year < 1398) and (publisher = 'Amirkabir')

(d)  $\Pi$  ( $\sigma$  (AUTHORS)  
AName  
City = 'Tehran')

SELECT AName  
FROM AUTHORS  
WHERE city = 'Tehran'

(e)  $\Pi$  ( $\sigma$  (STUDENTS)  
stName (Age > 25)  $\wedge$  (Field != 'computer Engineering')

SELECT stName  
FROM STUDENTS  
WHERE (Age > 25) and (Field != 'computer Engineering')

(f)  $\Pi$  ( $\sigma$  (STUDENTS, BORROWS, BOOKS)  
Books.DocId, Books.Title, Books.Publisher, Books.Year  
(STUDENTS.Field = 'Computer Engineering')  $\wedge$  (STUDENTS.stId = BORROWS.stId)  $\wedge$  (BORROWS.DocId = BOOKS.DocId)

(g)  $\Pi$  ( $\sigma$  (BOOKS  $\times$  HAS-WRITTEN)  
Books.Title (Books.DocId = HAS-WRITTEN.DocId)  $\wedge$  (HAS-WRITTEN.AName = 'Harrison')

(h)  $\Pi$  ( $\sigma$  (STUDENTS)  
Age = min(Age)  
stName, Age

SELECT Age, stName  
FROM STUDENTS  
ORDER BY Age ASC  
LIMIT(1)

(i)  $\Pi$  ( $\sigma$  (STUDENTS  $\times$  BORROWS  $\times$  HAS-WRITTEN)  
HAS-WRITTEN.AName (STUDENTS.stId = BORROWS.stId)  $\wedge$  (STUDENTS.stName = 'Carlos') and (HAS-WRITTEN.DocId = BORROWS.DocId)

(f):

: 4 (1<sup>st</sup> row 5 no 1) 4



```
SELECT Books.DocId, Books.Title, Books.Publisher, Books.Year
FROM STUDENTS, BORROWS, BOOKS
WHERE (STUDENTS.Field = 'Computer Engineering') and (STUDENTS.stId = BORROWS.stId)
and (BORROWS.DocId = Books.DocId)
```

(g):



```
SELECT Books.title
FROM Books, HAS-WRITTEN
WHERE (Books.DocId = HAS-WRITTEN.DocId) and (HAS-WRITTEN.AName = 'Harrison')
```

(h):



```
SELECT HAS-WRITTEN.AName
FROM STUDENTS, BORROWS, HAS-WRITTEN
WHERE (STUDENTS.stId = BORROWS.stId) and (STUDENTS.stName = 'carlos') and (HAS-WRITTEN.DocId =
BORROWS.DocId)
```

## Question 5)

(a)  $\pi ( \sigma ( \text{People} ) )$   
 NationalID BirthCity = 'Tehran' or BirthCity = 'Shiraz'



SELECT NationalID  
 FROM People  
 WHERE (BirthCity = 'Tehran') or  
 (BirthCity = 'Shiraz')

(b)  $\pi ( \sigma ( \text{People} ) )$   
 Name (BirthCity = 'Tabriz') ^ (BirthDate = 1377)



SELECT Name  
 FROM People  
 WHERE (BirthCity = 'Tabriz') and  
 (BirthDate = 1377)



(c)  $\Pi_{\text{NationalID}} ( \sigma_{\text{BirthDate} < 1377} ( \text{People} ) )$

6  
L → صفی بی

(d)  $\Pi_{\text{Name, BirthDate}} ( \sigma_{(\text{FatherID} = 1061426286) \text{ or } (\text{MotherID} = 1061426286)} ( \text{People} ) )$

L → صفی بی

(e)  $\Pi_{\text{s.Name}} ( \sigma_{\text{people.NationalID} = 6826241601 \wedge (\text{people.FatherID} = \text{s.fatherID}) \wedge (\text{people.MotherID} = \text{s.MotherID}) \wedge (\text{s.NationalID} \neq 6826241601)} ( \text{People} \times P_s(\text{People}) ) )$

L → صفی بی

(f)  $\Pi_{\text{People.NationalID}} ( \sigma_{\text{people.BirthCity} = \text{s.BirthCity} \wedge (\text{people.FatherID} = \text{s.NationalID}) \wedge (\text{people.MotherID} = \text{k.NationalID}) \wedge (\text{people.BirthCity} = \text{k.BirthCity})} ( \text{People} \times P_s(\text{People}) \times P_k(\text{People}) ) )$

L → صفی بی

(g)  $\Pi_{\text{people.Name, people.NationalID}} ( \sigma_{\text{people.BirthDate} > \text{s.BirthDate} \wedge (\text{people.FatherID} = \text{s.FatherID}) \wedge (\text{people.MotherID} = \text{s.MotherID})} ( \text{People} \times P_s(\text{People}) ) )$

L → صفی بی

(h)  $\Pi_{\text{P.Name, P.NationalID}} ( \sigma_{(\text{s.NationalID} \neq \text{P.NationalID}) \wedge (\text{s.FatherID} = \text{P.FatherID}) \wedge (\text{s.MotherID} = \text{P.MotherID}) \wedge (\text{people.FatherID} = \text{s.NationalID}) \wedge (\text{People.NationalID} = 1061426286)} ( \text{People} \times P_s(\text{People}) \times P_p(\text{People}) ) )$

L → صفی بی

(i)  $\Pi_{\text{People.Name}} ( \sigma_{(\text{people.NationalID} \neq \text{s.NationalID}) \wedge (\text{people.FatherID} = \text{s.FatherID}) \wedge (\text{people.MotherID} = \text{s.MotherID}) \wedge (\text{people.BirthDate} = \text{s.BirthDate})} ( \text{People} \times P_s(\text{People}) ) )$

(c):



```
SELECT NationalID
FROM People
WHERE (BirthDate < 1377)
```

(d):



```
SELECT Name, BirthDate
FROM People
WHERE (FatherID = 1061426286) or (MotherID = 1061426286)
```

(e):

```
SELECT S.Name
FROM People as P, People as S
WHERE (P.NationalID = 6826241601) and (P.FatherID = S.FatherID) and
(P.MotherID = S.MotherID) and (S.NationalID != 6826241601)
```

≠

(f):

SELECT P.NationalID  
FROM People as P, People as S, People as K  
WHERE (P.FatherID = S.NationalID) and (P.MotherID = K.NationalID)  
and (P.BirthCity = S.BirthCity) and (P.BirthCity = K.BirthCity)

(g):

SELECT P.Name, P.NationalID  
FROM People as P, People as S  
WHERE (P.BirthDate > S.BirthDate) and (P.FatherID = S.FatherID) and  
(P.MotherID = S.MotherID)

(h):

SELECT K.Name, K.NationalID  
FROM People as P, People as S, People as K  
WHERE (P.NationalID = 1061426286) and (P.FatherID = S.NationalID) and  
(S.MotherID = K.MotherID) and (S.FatherID = K.FatherID) and  
(S.NationalID != K.NationalID)

(K > S tuple  $\neq$  tuple)

(i):

SELECT P.Name  
FROM People as P, People as S  
WHERE (P.FatherID = S.FatherID) and (P.MotherID = S.MotherID) and  
(P.BirthDate = S.BirthDate) and (P.NationalID != S.NationalID)

(S > P tuple  $\neq$  tuple)



Question 6)

(a)

(b)

(c)

نام آینه هلیکام تا قبل از سال 2020 خریداری شده اند.  
نام مشتریانی که در سال 2020، خرید انجام داده اند.  
نام مشتریانی که تاکنون، خریدی انجام نداده اند.