

Subject:

Year. Month. Date. ( )

طرح سوال های انتقالی در ادامه:

$$r_1 \leftarrow \rho_{\text{person-name}} (\pi_{\text{manager-name}} (\text{manages})) \quad \text{فرد مدیران} \quad (1)$$

$$r_2 \leftarrow (r_1 \bowtie \text{employee}) \quad \text{مشخصات اول دستی ضمیمه مدیران}$$

$$r_3 \leftarrow \pi_{\text{person-name, city, street}} (\text{employee} \bowtie \text{works}) \quad \text{فرد ضمیمه کارکنان}$$

$$\text{result} \leftarrow r_3 \div r_2$$

این جواب سوالی است که اگر بخواهیم در جدول کارکنان نام مدیران را حذف کنیم و فقط نام مدیران را بماند و در جدول مدیران نام کارکنان را حذف کنیم و فقط نام کارکنان را بماند.

$$\pi_{\text{person-name}} \left\{ \left[ \rho_{\text{person-name}} \left( \left[ \rho_{\text{city, street}} \left( \text{employee} \bowtie \text{works} \right) \right] \bowtie \text{employee} \right) \right] \right\}$$

در اینجا  $\rho_{\text{city, street}}$  نام کارکنان را حذف می کند و فقط نام کارکنان را بماند.  $\rho_{\text{person-name}}$  نام مدیران را حذف می کند و فقط نام مدیران را بماند.

$$\pi_{\text{person-name}} \left( \left[ \left( \rho_{\text{company-name}} \left( \left[ \left( \rho_{\text{E-name} \rightarrow \text{person-name}} \left( \text{employee} \bowtie \text{works} \right) \right] \right) \right) \right] \right) \right] \quad (3)$$

count (company-name)  
as company\_count

$$\pi_{\text{person-name}} \left( \sigma_{\text{company-count} > 1} (r_1) \right)$$

Subject :

Year. Month. Date. ( )

در هر یک از اطلاعاتی توانایی وجود دارد که به سبب تغییرات اطلاعات در دست دارد سیستم مورد  
 در data برای مورد به سیستم باید این قوانین را رعایت کند  
 هر یک از اطلاعاتی باید مقدارش غیر صفر باشد. اگر سبب باشد که در صورت صحت جابجایی غیر صحیح باشد  
 در صورتی که candidate key باشد این باید غیر صفر باشد و غیره

**فصل چهارم SQL : structured Query Language**

سیستم مدیریت پایگاه داده (DBMS) : مجموعه‌ای نرم‌افزاری که برای ذخیره و بازیابی اطلاعات و سازماندهی  
 بانک‌های اطلاعاتی طراحی رایج شده‌اند که جمله‌ای از مجموعه‌های داده تشکیل داده‌اند. هر کدام از این بخش‌ها  
 نوعی یکی از اعمال مختلف برای آن می‌باشد

زبان‌های اطلاعاتی : یک زبان ساده برای ایجاد و اصلاح و استفاده از بانک‌های اطلاعاتی طراحی شده‌اند  
 زبان‌های سطح چهارم : **4GL** یا **Fourth generation lang** بخشی از DBMS که برای طراحی نرم‌ها  
 خاص برای مورد و نمایش اطلاعات استفاده می‌شود  
 زبان‌های سطح پنجم : **5GL** یا **embedded lang** به این وسیله می‌تواند دستورات زبان‌های سطح چهارم را  
 یکدیگر را در داخل یکی از زبان‌های سطح سوم (مانند C++) استفاده نمود

زبان‌های سطح ششم : **6GL** یا **intelligent lang** : همانند زبان‌های سطح چهارم و پنجم برای بانک‌های اطلاعاتی  
 طراحی شده‌اند. از آنجا که SQL به عنوان زبان استاندارد در محیط‌های سطح چهارم اطلاعاتی استفاده شده  
 است، در این فصل به تشریح این زبان می‌پردازیم

تفاوت زبان‌های سطح ششم در DBMS بر اساس سه ویژگی است : اطلاعاتی طراحی شده است  
 (2) ترکیب (Syntax) : برای تعیین سندهای بخش بر اساس ضوابطی از SQL است. استاندارد می‌باشد

Subject :

Year . Month . Date . ( )

تاریخچه: توسعه IBM

1970 IBM در این سال

1986 SQL-86 for ISO و ANSI

1989 IBM SAA-SQL

1989 SQL-89 for ANSI

1992 SQL-92 for ANSI

1999 SQL-99

زبان SQL دارای چند بخش می باشد

DDL : Data Definition Language

DML : interactive data-manipulation Lang

تعریف زبان جدولی جو در اساس جدولی و رابطه های جدولی می باشد

view Definition

کشی از DDL SQL در سطح جدولی دستور برای تعریف view می باشد

Transaction Control

حالی دستور برای کنترل شروع و خاتمه یک تراکنش می باشد

Embedded and Dynamic SQL

تقریباً می تواند چون جملات SQL می تواند در زبان های برنامه نویسی هم استفاده شود + C++ کاربرد

Integrity

درستی یا نادرستی اظہار می باشد و دستوراتی که در DDL SQL است که دارای دستوراتی

برای تعیین صحت داده های جدولی می باشد که باید در جدول داده رعایت شود

Subject:

Year. Month. Date. ( )

نوع SQL برای چنین چیزی چیست؟

Authorization

کمی از SQL DDL به جدولی دسترسی برای تعیین اجازه دسترسی به رابطه ها و view های جدید

27.9.2

دسته having ← شرط چهارم یک group است یعنی اند

در بیان خودی group by شرط را اجرا کند

where عملی کند

بخصوص null

در یک رابطه می توانیم مقدار null داشته باشیم

null را شناخته باشد

✓ یا اصلاً وجود نداشته باشد

در null و در جدولی که مقدارش unknown می شود

unknown مقادیر نسبت به True و False

در جدولی که اگر در تمام جمع به از حالت count مقدار null در برقیه حالت ها ignore می شود

در در یک ستون هم مقادیر null باشند خودی این روابط جمع null حالت است برای count برقرار نیست

عملگر in ← عملگر متعلق بودن حتی در این محورها باشد

B-A=0

# 7370  
75

Subject:

Year. Month. Date. ( )

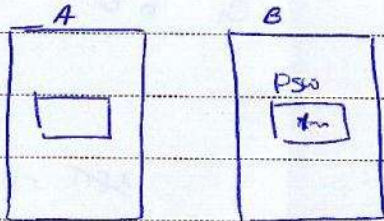
دستی‌اندازنده‌ها، اعلا فستورده‌ها، پادیه‌ها، صلا‌ای‌اندازنده‌ها؟

Task type f.e. is

A, B, C, D, E T1;

همین‌طور Task‌ها چگونه‌ها می‌باشند؟

مثال: اگر برنامه A یک‌بار برای برنامه B اجرا شود، پس برای B مقداری کار انجام می‌دهد. B می‌تواند



در این PSW بود

در PSW برای B یک بیت خاص Set می‌شود، B در رجوع به PSW متوجه وقفه می‌شود و Interrupt Handler را فعال می‌کند پس از handle شدن وقفه، اجرای کاری از کبر وقفه می‌شود

Subject:

Year. Month. Date. ( )

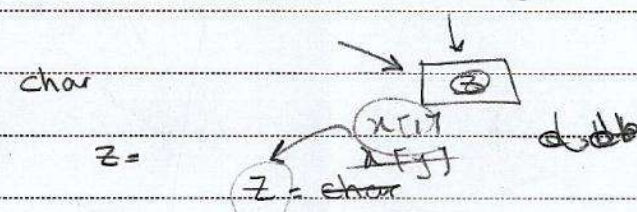
$y = 1$   
 $x = 0$   
 $z = 'd'$   
 $x[2] = x[0]$   
 a b b

$f(x[0] = b)$

$y = 1$   
 $z = 'd'$   
 a b b

$y = 1$   
 $y = 1$   
 $x[0] = 'd'$   
 $x[2] = x[1]$   
 a b b

$z = b$   
 $z = b$   
 $z = d$   
 $x[2] = b$   
 $x[1] = d$



$z \rightarrow x[1]$

Subject:

Date:

programming language Course  
 homework assignment mid term exam solution search as  
 Sample exam  
 binding orthogonality  
 Fixed exam  
 Filetype: pdf  
 word

real  $x$  Dynamic Scope  
 integer Static

Program main:

var x, y: integer;

Procedure R:

var z: real;

begin

z = x + 1;

end R;

Procedure Q:

var z: real;

begin

R;

end Q;

Procedure P:

var x: Boolean;

begin

Q;

end P;

begin main,

P

end;



Subject: \_\_\_\_\_

Date: \_\_\_\_\_

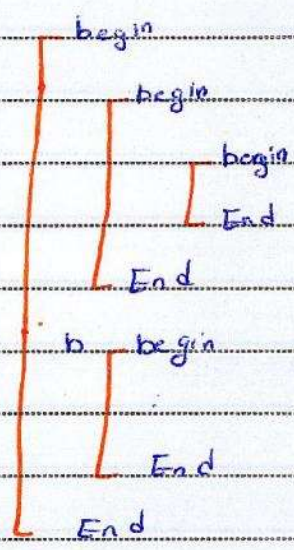
دینامیک اسکوپ (Dynamic Scope) : APL, SNOBOL, LISP  
 ایستاتیک اسکوپ (Static Scope) : PL/I, Fortran, Pascal, C

Dynamic Scope: جستجو در تمام اسکوپ‌ها  
 Static Scope: جستجو در اسکوپ محلی و اسکوپ‌های پدری

Block Structure: اسکوپ‌ها در بلوک‌ها تعریف می‌شوند

**Block Structure (SNOBOL):**

end, begin  
 procedure



**Block Structure (C) and Scope**

- ① جستجو در اسکوپ محلی اولویت دارد
  - ② اگر در اسکوپ محلی پیدا نشود، به اسکوپ پدری می‌رود
- Most Closely Nested

Subject:

Encapsulation

Data:

- ③ اگر از داخل یک بکند اعلام ننماید بود هرگز برای آن متغیر آن بکند صدای داخلی تر جستجوی ندارد.
- ④ هر بکند جز محیط خودش نسبتاً بکند جزئی است یعنی نسبتاً در آن را اعلام کرده

Current Environment Table

تقریبی را به عنوان یک جدول در نظر بگیرید

Procedure SUB (x: integer)

var y: real;

Z: array [1..3] of

Procedure SUB2

!

end;

begin

!

End

x	تقریبی عدد integer
y	real
Z	Description
SUB2	code beg SUB2 است

activation record جدولی است از

Block Structure Using Scope

87, 9, 10

Program main( )

Procedure Sub( )

Var t = 100

begin

print (t)

t = t + 1

end;

begin

sub(5);

sub(7);

end;

بند جدیدی اضافه کردن. اجزای مختلف:

عملی است. در هر خطی که در هر خطی که در هر خطی که

حفظ می شود

حذف  
تعدادی

100 100

101 101

دو مورد در حافظه ماندگار می باشد. اجزای مختلف

تعدادی (Retention) Fortran, Cobol

حذف (Deletion) C, Pascal, Ada

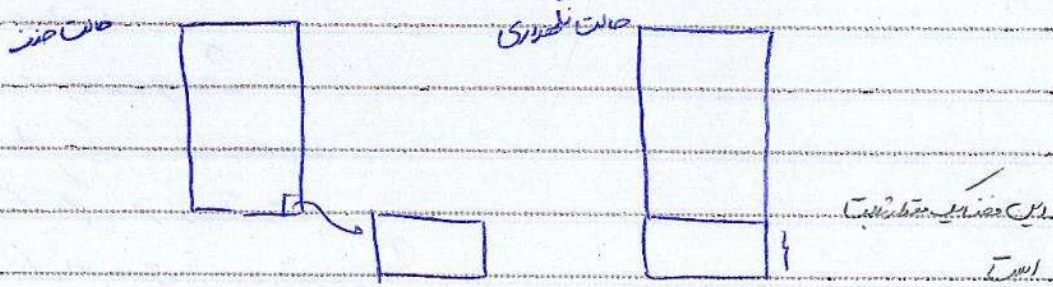
بسیار کم است

در حالت حذف، جدول محیط محلی جدول Code Segment نام دارد

در حالت نگهداری، جدول محیط محلی در جدول Code Segment قرار می گیرد

← اگر برای اجزای ماندگار، حافظه با سیستم کامپیوتر مناسب است

← در جدول اجزای ماندگار، فضای بیشتری می گیرد. تعدادی جدول هر دو برنامه اجزای ماندگار و ماندگار است



Subject:

Data:

در صورتی که نامی از نامهای پیشین در دسترس نباشد

کلاسها (class) از طریق (class) استفاده میکنند

کلاسها میتوانند از کلاسهای دیگر ارث بردارند (class)

اجزای کلاسها: actual

نوع داده Scope: در سیستم متغیرها را در یک محدوده تعریف میکند

class: کلاسها میتوانند از کلاسهای دیگر ارث بردارند (public static)

extern: در ابتدای یک فایل module تعریف میشود

استثنا: دادهها باید با یکدیگر سازگار باشند

main ( )

نوع دادههای اصلی

procedure sub(x, y)

متغیرها در یک محدوده تعریف میشوند

begin

جملهها در یک خط هستند (z = a, y = z)

end

کلاسها در Ada به نامهای class نامیده میشوند

begin

sub(y = z, x = a)

sub(a, z)

end

نوع دادههای اصلی: integer, real, boolean, character

✓

✓

✓

expression

✓

✓

Subject:

Year. Month. Date. ( )

### Dependency Preserving

result =  $\alpha$   
تفاوتی result تعیین کن

برای هر FD  $\alpha \rightarrow \beta$  در  $F$

for each  $R_i$

این result بر  $R_i$  درست است

$\alpha \rightarrow \beta$  در  $R_i$

این شرط درست است

result = result  $\cup$   $R_i$

### BCNF

برای  $R$  اگر  $F$  داشته باشیم و  $F^+$  را حساب کنیم  
decomposition:  $F^+$  استفاده کنیم

راه دیگر: برای هر  $\alpha \rightarrow \beta$  در  $F^+$  و  $\alpha \subset R_i$  و  $\alpha \rightarrow \beta$  در  $R_i$  است

صحیح است.  $R_i$  را با  $\alpha$  و  $\beta$  در  $R_i$  است BCNF است

### BCNF Decomposition

result =  $\{R\}$

done = false

compute  $F^+$

while (not done) do

if (there is a schema  $R_i$  in result that is not in BCNF)

then begin

let  $\alpha \rightarrow \beta$  nontrivial FD that holds on  $R_i$

such that  $\alpha \rightarrow \beta$  is not in  $F^+$ , and  $\alpha \cap \beta = \emptyset$

result = (result -  $R_i$ )  $\cup$  ( $R_i \rightarrow \beta$ )  $\cup$  ( $\alpha, \beta$ )

end

Subject:

Year. Month. Date. ( )

3NF

Decomposition

$F_c$

$i := 0$

For each functional dependency  $\alpha \rightarrow \beta$  in  $F_c$  do

if none of the schemas  $R_j, j=1, \dots, i$  contains  $\alpha\beta$

then begin

$i := i + 1$

$R_i = \alpha\beta$

if none of the schemas  $R_j, j=1, 2, \dots, i$  contains a candidate key for  $R_i$

then begin

$i := i + 1$

$R_i =$  any candidate key for  $R_i$

end

result :=  $\{R_i\}$

done := false

compute  $D^+$

~~while~~ ( 4NF )

if ( )

then begin

let  $\alpha \twoheadrightarrow \beta$  be a nontrivial multivalued dependency that holds on  $R_i$  such that  $\alpha \twoheadrightarrow R_i$  is not in  $D^+$  and  $\alpha \cap \beta = \emptyset$

result := (result -  $R_i$ )  $\cup$  ( $R_i - \beta$ )  $\cup$  ( $\alpha, \beta$ )

Subject:

Year. Month. Date. ( )

نام درس

AUT-PL-1 عنوان این F-mail در این  
"Programming Language Course Syllabi" گوگل در search

تعمیرات این کلاس

این کلاس برای یادگیری بیشتر در مورد استفاده از این زبان

مباحث این کلاس

عنوان این کلاس

1. روش کار این زبان و نحوه کار با این زبان

3. هر چه بتوانیم ابزارهای قدرتمندی برای این زبان پیدا کنیم

Fortran

بسیار قدیمی و سنگین

Cobol

Syntax آن خیلی پیچیده است و کار با آن سخت است

Pascal

یادگیری آن بسیار آسان است

2. از همین برای استفاده از این زبان

تاریخچه

این زبان در سال 1957 ساخته شد

این زبان در سال 1960 توسعه پیدا کرد

IDE یکی از ابزارهای رایج برای کار با این زبان است

این کلاس

عنوان این کلاس

این زبان Function خاصیت دارد

Lisp, Haskell

Subject:

Year. Month. Date. ( )

نوعی از سریال‌های استیجی

در این جا Sequence نام داریم یعنی Rule یی

Father (Ali, Reza)

PSF. خصوصی کنیم

Father (Ali, Kazem)

Father (Reza, Bahram)

رابطه (Rule) را ترتیب می‌دهیم و جواب می‌دهیم

Grand\_Father (xy) :- Father (x, z), Father (z, y)

YACC یک سیستم برسی نحوی compiler

نوعی از سریال‌های استیجی

script ← یک feature که در سریال‌های زبان را دارد

Both Files ← هم برای از دست دادن را در خود دارد

Python ← برای سریال‌های استیجی

Mark-up Language

زبان نشانه‌گذاری

Internet Explorer که توسط مایکروسافت ساخته شده برای HTML (سایت)

html → صفحات را مشخص می‌کند و مشخصات page را می‌دهد

کامپیوتر به سریال‌ها page به هم وصل می‌کند از JS استفاده می‌کنند در CSS (cascading style sheet)

یک سیستم ترتیب‌دهی مشخصات است



Subject:

Year. Month. Date. ( )

خوبه انجام: 7, 19

تقریباً (Syntax)

Syntax: ترتیبی است. قواعد مشخصی دارد. لغات برنامه نویسی را میسر می کند. ترتیبی بودن کدها باعث می شود که قابل فهم باشد.

محدودیت زبان

ASCII 8bit

0-127 →

حرف انگلیسی

128-255 →

زبان توسعه یافته (extended)

محدودترها

این برای رفع مشکل برای حالت است. نیاز دارد برای عملیات خاص. بعضی کلمات از طریق سیستم های دیگر

unicode

16bit

شماره ها (identifier) ترکیبی از حرف و اعداد برای نامگذاری

کلمات اضافی (noise words) (مانند و خاصه noise) (موجود است)

از طریق قابلیت ارتباط بین زبان

نوع داده ها

main frame رده های از server همان IRM تولید می کنند. Time sharing برای برسان

به سیستمی که وصل است به سیستم های دیگر

نوع داده ها زیاد است. اگر بتوانیم در هر جایی که می خواهیم به هم وصل شویم. در صورتی که می خواهیم به هم وصل شویم. Free

در زمینه assembly، قیمت free نداریم بلکه قیمت آزاد است.

تولید کننده کدها

خوانایی، قابلیت توسعه پذیری، قابلیت ترجمه، هم وجود دارد

از زبان خوب که می توانیم راحت است و کارها خوب می آید

Computers راحت تر است و میزان خطا کمتر است