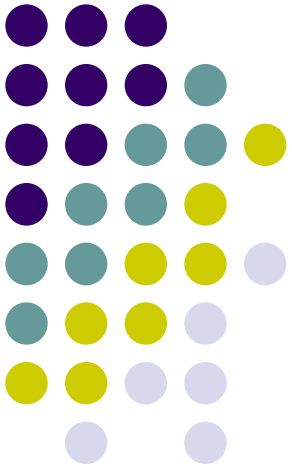
Applied Faculty
Hama University

Database 2

قواعد البيانات الغرضية – العلائقية

أنماط البيانات المجردة

المصفوفات المتغيرة



قواعد البيانات الغرضية – العلائقية

- امتداد لقواعد البيانات العلائقية
- مجموعة من الآليات الموجودة في العديد من نظم إدارة قواعد البيانات المعاصرة

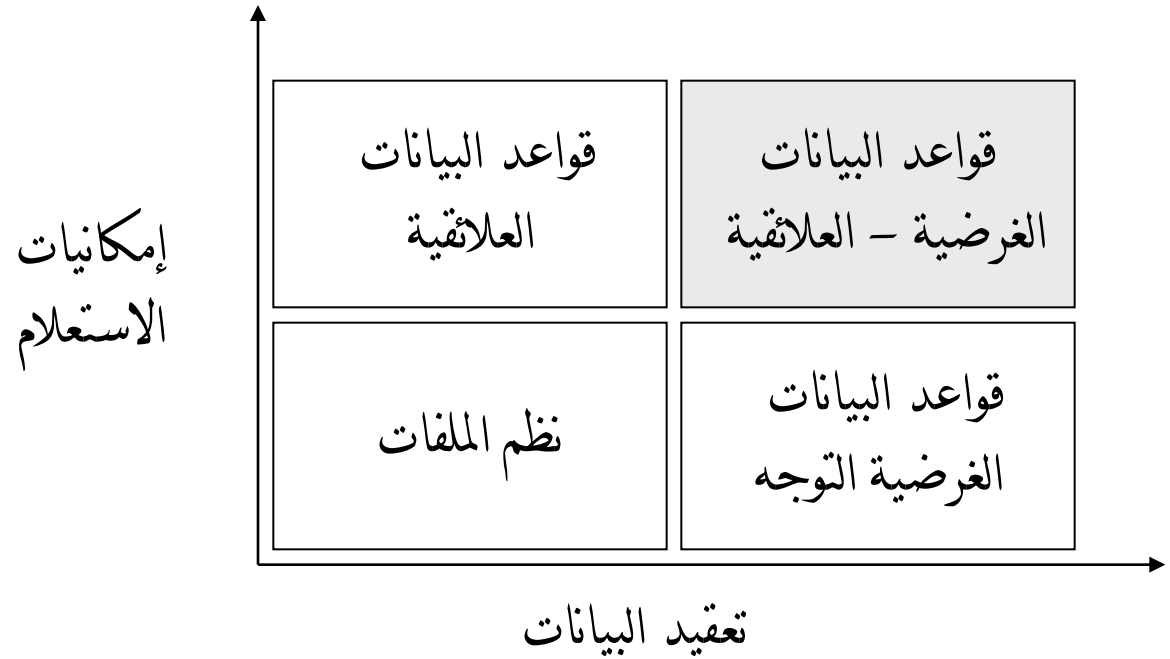
- تعريف البيانات
  - أنماط البيانات المجردة
  - المصفوفات المتغيرة
  - العلاقات المتداخلة
- معالجة البيانات
  - الإجرائيات المخزنة
  - القوادح

قواعد البيانات الغرضية – العلائقية

- النموذج العلائقي
  - يستند إلى بنى معطيات "بسيطة" (الجداول)
  - يعتمد على القيم المحتواة في الجداول

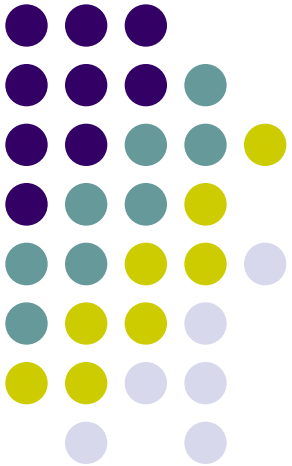
- النموذج العلائقي – الغرضي
  - يستند إلى إمكانية تعريف بنى معطيات معقدة
  - مفهوم الجدول موسع ليشمل إمكانية تعريف أنماط بيانات مجردة

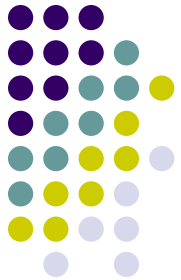
قواعد البيانات الغرضية – العلائقية



أنماط البيانات المجردة

Abstract Data Types





أنماط البيانات المجردة

- نمط البيانات المجرد (Abstract DataType)
  - هو بنية معطيات معرفة من قبل المستخدم و مشتركة بين عدة جداول (أو عدة أنماط) .

- تحتوي الأنماط المجردة على طرائق (methods) تسمح بمعالجة أغراض النمط المجرد .

إنشاء نمط مجرد

```
CREATE TYPE ADDRESS_TY AS OBJECT
(
  street VARCHAR2(50),
  city   VARCHAR2(25),
  state  CHAR(2),
  zip    NUMBER
)
/
```

- النمط ليس جدولاً !
  - لا يمكن إدخال البيانات إليه مباشرة

استخدام النمط في تعريف جدول

```
CREATE TABLE PERSON (
  name      VARCHAR2(25),
  address   ADDRESS_TY
);
```

PERSON				
name	address			
	street	city	state	zip

استخدام النمط في تعريف نمط آخر

```
CREATE TYPE PERSON_TY AS OBJECT
(
  name      VARCHAR2(25),
  address   ADDRESS_TY
)
/
```

PERSON_TY				
name	address			
	street	city	state	zip

إنشاء جدول من نمط

```
CREATE TABLE PERSON OF PERSON_TY;
```

PERSON				
name	address			
	street	city	state	zip

≡

```
CREATE TABLE PERSON (
  name      VARCHAR2(25),
  address   ADDRESS_TY
);
```

استخدام نمط في إنشاء جدول

```
CREATE TABLE CUSTOMER(
  customer_ID  NUMBER,
  person       PERSON_TY
);
```

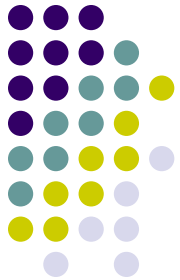
CUSTOMER					
customer_ID	person				
	name	address			
		street	city	state	zip

استخدام النمط في إنشاء جدول

```
DESCRIBE CUSTOMER
```

Name	Null?	Type
CUSTOMER_ID		NUMBER
PERSON		PERSON_TY

ايجاد أعمدة جدول من قاموس البيانات

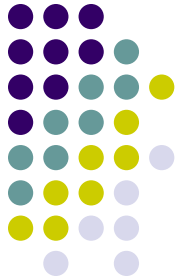


```
SELECT column_name, data_type
  FROM USER_TAB_COLUMNS
 WHERE table_name = 'CUSTOMER';
```

قاموس البيانات
(أعمدة جداول المستخدم)

COLUMN_NAME	DATA_TYPE
CUSTOMER_ID	NUMBER
PERSON	PERSON_TY

ايجاد واصفات نمط من قاموس البيانات

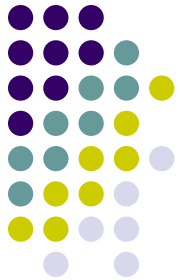


```
SELECT attr_name, length, attr_type_name
  FROM USER_TYPE_ATTRS
 WHERE type_name = 'PERSON_TY';
```

قاموس البيانات
(واصفات أنماط المستخدم)

ATTR_NAME	LENGTH	ATTR_TYPE_NAME
NAME	25	VARCHAR2
ADDRESS		ADDRESS_TY

ايجاد واصفات نمط من قاموس البيانات



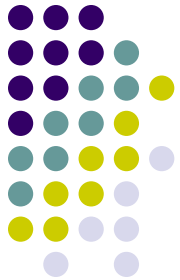
```
SELECT attr_name, length, attr_type_name
  FROM USER_TYPE_ATTRS
 WHERE type_name = 'ADDRESS_TY';
```

ATTR_NAME	LENGTH	ATTR_TYPE_NAME
STREET	50	VARCHAR2
CITY	25	VARCHAR2
STATE	2	CHAR
ZIP		NUMBER

قاموس البيانات

معرفة من قبل المستخدم	USER_TABLES	جداول المستخدم
	USER_TYPES	أنماط المستخدم
	USER_TAB_COLUMNS	أعمدة جداول المستخدم
	USER_TYPE_ATTRS	واصفات أنماط المستخدم
	USER_CONSTRAINTS	قيود المستخدم
	USER_CONS_COLUMNS	أعمدة قيود المستخدم
معرفة من قبل المستخدم + يملك ترخيص الوصول إليها	ALL_TABLES	جميع الجداول
	ALL_TYPES	جميع الأنماط
	ALL_TAB_COLUMNS	أعمدة جميع الجداول
	ALL_TYPE_ATTRS	واصفات جميع الأنماط
	ALL_CONSTRAINTS	جميع القيود
	ALL_CONS_COLUMNS	أعمدة جميع القيود
	... ...	... ...

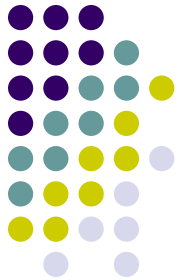
إدراج البيانات في عمود نمط مجرد



```
INSERT INTO CUSTOMER VALUES
(1,
  PERSON_TY('John Smith',
    ADDRESS_TY('57 PLEASANET ST', 'FINN','NH',11111)));
```

```
INSERT INTO CUSTOMER VALUES
(2,
  PERSON_TY('Seymour Hester',
    ADDRESS_TY('1 STEPAHEAD RD', 'BRIANT','NH',11111)));
```

استخراج البيانات من نمط مجرد



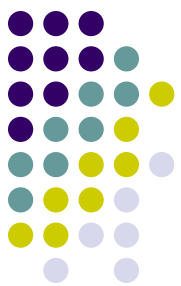
```
SELECT customer_ID
  FROM CUSTOMER;
```

CUSTOMER_ID
1
2

```
SELECT *
  FROM CUSTOMER;
```

CUSTOMER_ID	PERSON(NAME, ADDRESS(STREET,CITY,STATE,ZIP))
1	PERSON_TY('John Smith',ADDRESS_TY('57 PLEASANET ST','FINN','NH',11111))
2	PERSON_TY('Seymour Hester',ADDRESS_TY('1 STEPAHEAD RD','BRIANT','NH',11111))

استخراج البيانات من نمط مجرد

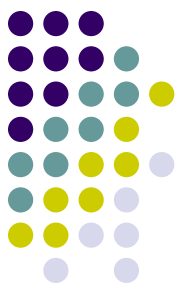


```
SELECT Name
  FROM CUSTOMER;
```

```
ERROR : invalid column name
```

- Name ليس اسم عمود في الجدول Customer

استخراج البيانات من نمط مجرد

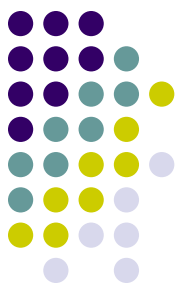


```
SELECT C.Customer_ID, C.Person.Name
  FROM CUSTOMER C;
```

CUSTOMER_ID	PERSON.NAME
1	John Smith
2	Seymour Hester

Table . Column . Attribute

استخراج البيانات من نمط مجرد

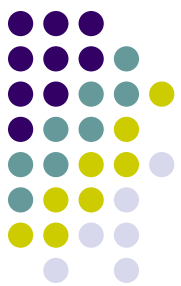


```
SELECT C.Person.Address.Street
  FROM CUSTOMER C;
```

PERSON.ADDRESS.STREET
57 PLEASANET ST
1 STEPAHEAD RD

Table . Column . Column . Attribute

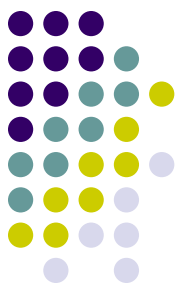
استخراج البيانات من نمط مجرد



```
SELECT C.Person.Name, C.Person.Address.City
  FROM CUSTOMER C
 WHERE C.Person.Address.City LIKE '%F%';
```

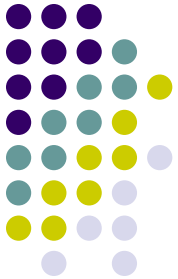
PERSON.NAME	PERSON.ADDRESS.CITY
John Smith	FINN

تعديل البيانات في نمط مجرد



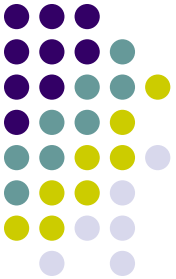
```
UPDATE CUSTOMER C
   SET C.Person.Address.City = 'HART'
 WHERE C.Person.Address.City = 'BRIANT';
```

Methods – الطرائق
التصريح عن الطرائق في النمط



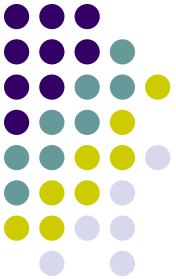
```
CREATE OR REPLACE TYPE person_type AS OBJECT
(
  first_name VARCHAR2 (30),
  last_name VARCHAR2 (30),
  phone_number VARCHAR2(20),
  date_of_birth DATE,
  MEMBER FUNCTION name_and_phone RETURN VARCHAR2,
  MEMBER FUNCTION ageOnDate(onDate DATE) RETURN INTEGER
)
/
```

Methods – الطرائق
تعريف الطرائق في جسم النمط



```
CREATE OR REPLACE TYPE BODY person_type AS

MEMBER FUNCTION name_and_phone RETURN VARCHAR2 IS
BEGIN
  RETURN (SELF.first_name||' '||SELF.last_name||' Phone: '
     ||SELF.phone_number);
END name_and_phone;

MEMBER FUNCTION ageOnDate(onDate DATE) RETURN INTEGER IS
BEGIN
  RETURN MONTHS_BETWEEN(onDate,SELF.date_of_birth)/12;
END ageOnDate;

END;
/
```

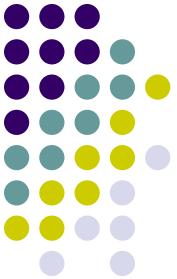
Methods – الطرائق



```
CREATE TABLE person OF person_type;
```

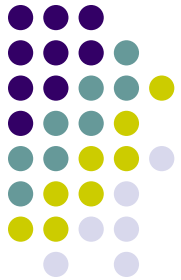
```
INSERT INTO person VALUES
('Robert', 'Freeman', '904.200.2020', to_date('01-JUN-1979'));
INSERT INTO person VALUES
('Deborah', 'Freeman', '014.200.2020', to_date('01-MAY-1986'));
```

Methods – الطرائق
استدعاء الطرائق



```
SELECT p.name_and_phone(), p.ageOnDate(current_date)
FROM person p;
```

P.NAME_AND_PHONE()	P.AGEONDATE(CURRENT_DATE)
Robert Freeman Phone: 904.200.2020	31
Deborah Freeman Phone: 904.200.2020	24

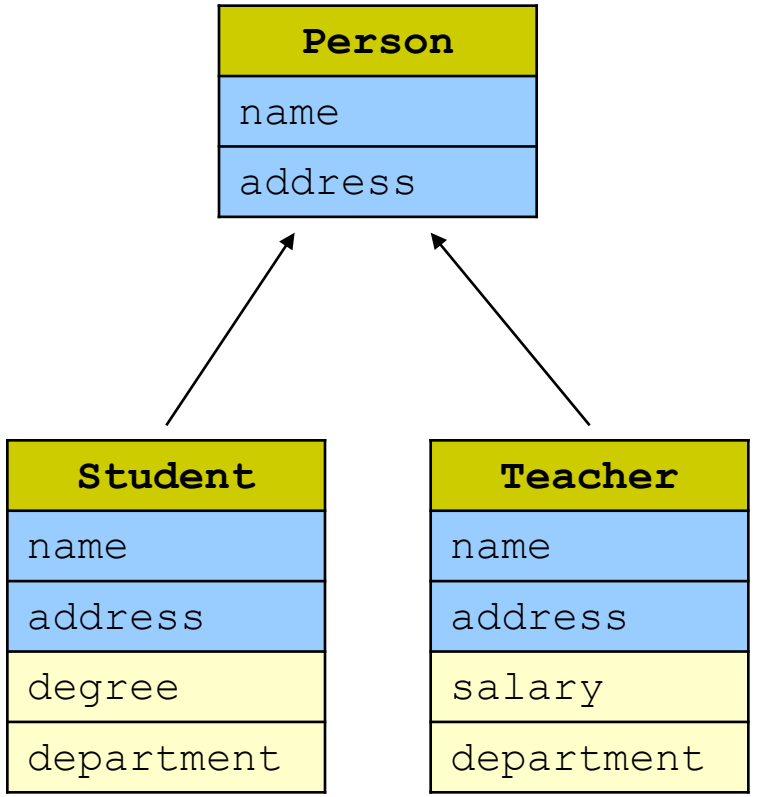


الوراثة

- يمكن إنشاء نمط كنمط جزئي من نمط موجود
- فقط الوراثة المفردة ممكنة
  - النمط الجزئي (الابن) يمكن اشتقاقه من نمط أب واحد فقط
  - الوراثة المتعددة غير ممكنة
- يرث النمط الابن جميع واصفات و طرائق النمط الأب المباشر
- يمكن للنمط الابن إضافة واصفات و طرائق جديدة و يمكنه إعادة كتابة الطرائق الموروثة

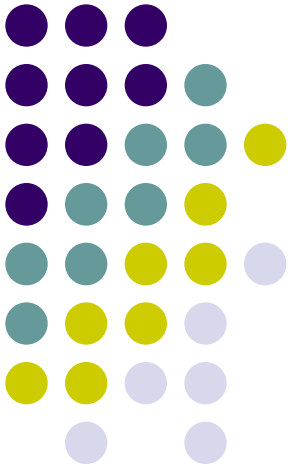
الوراثة

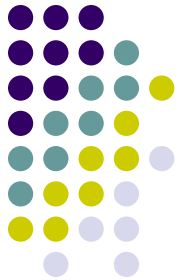
```
CREATE TYPE Person AS OBJECT (
  name     VARCHAR2(20),
  address  VARCHAR2(20)
) NOT FINAL
/
CREATE TYPE Student UNDER Person (
  degree      VARCHAR2(20),
  department  VARCHAR2(20)
)
/
CREATE TYPE Teacher UNDER Person (
  salary      INTEGER,
  department  VARCHAR2(20)
)
/
```



المجمعات

Collections





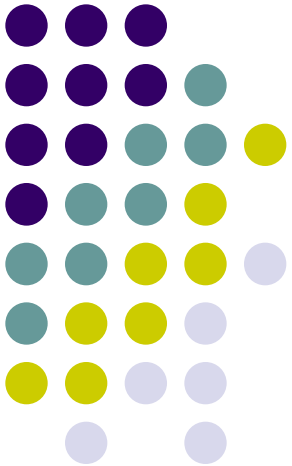
المجمعات

- يسمح المجمع بتخزين البيانات المتكررة (أكثر من غرض) في نفس السطر
  - إلغاء تقييس الجدول (de-normalization)
  - Not-First Normal Form (NF2)

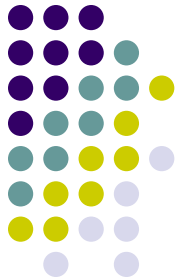
- تدعم Oracle نوعين من المجمعات
  - المصفوفات المتغيرة (varying arrays)
  - الجداول المتداخلة (nested tables)

المصفوفات المتغيرة

Varying Arrays



المصفوفات المتغيرة

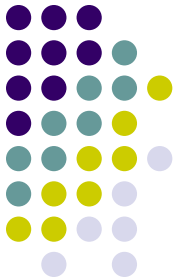


```
CREATE TABLE Borrower (
  name VARCHAR2(25),
  tool VARCHAR2(25),
  CONSTRAINT borrower_pk PRIMARY KEY (name, tool)
);
```

BORROWER	
NAME	TOOL
Jed Hopkins	Shovel
Jed Hopkins	Axe
Jed Hopkins	Hammer
John Smith	Shovel
John Smith	Saw

BORROWER	
NAME	TOOL
Jed Hopkins	{Shovel, Axe, Hammer}
John Smith	{Shovel, Saw}

تعريف مصفوفة متغيرة

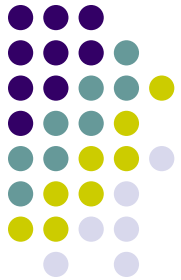


BORROWER	
NAME	TOOL
Jed Hopkins	{Shovel, Axe, Hammer}
John Smith	{Shovel, Saw}

```
CREATE OR REPLACE TYPE TOOLS_VA
    AS VARRAY(5) OF VARCHAR2(25);
```

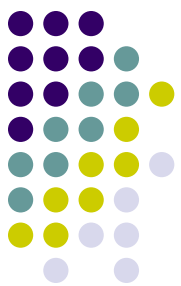
```
CREATE TABLE Borrower (
  name VARCHAR2(25) PRIMARY KEY ,
  tools TOOLS_VA
);
```

إدخال البيانات في مصفوفة متغيرة



```
INSERT INTO BORROWER VALUES
('Jed Hopkins', TOOLS_VA('Shovel', 'Axe', 'Hammer'));
INSERT INTO BORROWER VALUES
('John Smith', TOOLS_VA('Shovel', 'Saw'));
```

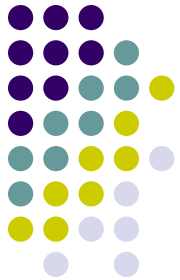
BORROWER	
NAME	TOOLS
Jed Hopkins	{Shovel, Axe, Hammer}
John Smith	{Shovel, Saw}



إدخال البيانات في مصفوفة متغيرة

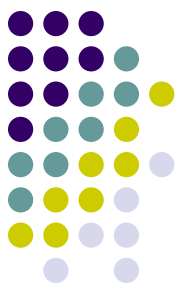
```
SQL> SELECT * FROM Borrower;

NAME           TOOLS
-------------- ------------------------------------
Jed Hopkins    TOOLS_VA('Shovel', 'Axe', 'Hammer')
John Smith     TOOLS_VA('Shovel', 'Saw')
```

استخراج المعلومات من مصفوفة متغيرة



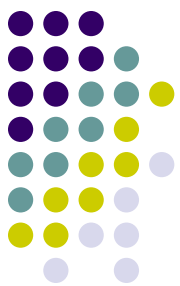
```
DECLARE
  CURSOR borrower_cursor IS
    SELECT * FROM borrower;
  borrower_rec borrower_cursor%ROWTYPE;
BEGIN
  FOR borrower_rec IN borrower_cursor LOOP
    DBMS_OUTPUT.PUT_LINE('Borrower Name: '||borrower_rec.name);
    FOR I IN 1..borrower_rec.tools.COUNT LOOP
      DBMS_OUTPUT.PUT_LINE(borrower_rec.tools(I));
    END LOOP;
  END LOOP;
END;
/
```

استخراج المعلومات من مصفوفة متغيرة



```
Borrower Name: Jed Hopkins
Shovel
Axe
Hammer
Borrower Name: John Smith
Shovel
Saw
```

طرائق الوصول إلى المصفوفات المتغيرة في PL/SQL



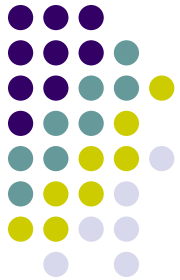
Method	Description
COUNT	Returns the number of elements in the collection
DELETE	Causes the removal of one, all, or a given range of elements in a VARRAY
EXISTS	Returns TRUE if the element exists
EXTEND	Adds a given number of elements to the VARRAY
FIRST	Returns the first index in the VARRAY
LAST	Returns the last index in the VARRAY
LIMIT	Returns the maximum number of elements in the VARRAY
NEXT	Returns the next index number in the VARRAY
PRIOR	Returns the previous index number
TRIM	Removes one or more elements from a collection

تعديل المصفوفات المتغيرة

تعديل سطر بكامله

```
UPDATE borrower
SET tools = TOOLS_VA('Shovel', 'Axe')
WHERE name = 'Jed Hopkins';
```

تعديل المصفوفات المتغيرة



تعديل عناصر مفردة من المصفوفة

```
DECLARE
  var_tools_va TOOLS_VA;
BEGIN
  SELECT tools INTO var_tools_va
  FROM borrower
  WHERE name = 'Jed Hopkins';
-- Manipulate the VARRAY.
  ...
  ...
-- Now, update the object itself
  UPDATE borrower SET tools = var_tools_va
  WHERE name = 'Jed Hopkins';
END;
/
```

قراءة المصفوفة إلى متحول PL/SQL

معالجة المتحول

تحديث الجدول و وضع قيمة المتحول في مكانها الأصلي

تعديل المصفوفات المتغيرة

تعديل عناصر مفردة من المصفوفة

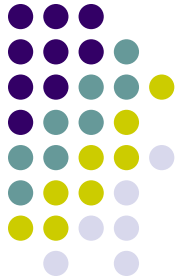
```
-- Add to the VARRAY.
  var_tools_va.EXTEND;
  var_tools_va(var_tools_va.LAST) := 'Saw';
```

إضافة عنصر جديد

```
-- Update specific elements
  var_tools_va(1) := 'Axe';
  var_tools_va(2) := 'Shovel';
```

تعديل عناصر محددة

تعديل المصفوفات المتغيرة



تعديل عناصر مفردة من المصفوفة

```
DECLARE
  var_tools_va TOOLS_VA;
BEGIN
  SELECT tools
  INTO var_tools_va
  FROM borrower
  WHERE name = 'Jed Hopkins';
-- Add to the VARRAY.
  var_tools_va.EXTEND;
  var_tools_va(var_tools_va.LAST) := 'Saw';
-- Update specific elements
  var_tools_va(1) := 'Axe';
  var_tools_va(2) := 'Shovel';
-- Now, update the object itself
  UPDATE borrower SET tools = var_tools_va
  WHERE name = 'Jed Hopkins';
END;
/
```