

وزارة التعليم العالي والبحث العلمي

كلية الحدااء الجامعة

علم الحاسوب

المرحلة الثالثة



ملخص عن الإجراءات المخزنة

STORED PROCEDURE

في لغة SQL

بأشراف الأستاذ :

ماهر خلف حسين

اعداد الطالبة :

رحمة لبيب الطائي



Stored Procedure

2019-2020

## Stored procedures

هي عبارة عن أوامر نكتبها بلغة **sql** في قاعدة البيانات ونعطيها اسم معين ثم نقوم بتنفيذ هذه الإجراءات المكتوبة على السيرفر لغرض القيام او الحصول على عملية معينة .

**مثلا** يمكننا استخدام **stored procedure** (الإجراءات المخزنة) لغرض جلب البيانات ثم نحصل على نتيجة في البرنامج.

وأيضا نستخدم الإجراءات المخزنة لعملية إضافة بيانات الى جدول ما او نقوم بحذف البيانات من جدول معين وأيضا نقوم بعملية تعديل البيانات من جدل معين.

**ما الغرض من stored procedure اذا كانت تقوم بنفس العمل الذي قمنا به سابقا بواسطة php&mysql ؟**

ان استخدامها يعزز البرنامج وانها تساهم في رفع أداء البرنامج لانها تجعل الاكواد المتعلقة بالسيرفر تنفذ على مستوى السيرفر وليس على مستوى البرنامج وهذا يضمن اشتغال البرنامج بسرعة وبأداء عالي **هذا من جهة.**

**ومن جهة أخرى:** فهي تجعل البرنامج سهل التعديل وسهل الصيانة لانه سوف يحتوي على الاكواد المتعلقة بلغة البرمجة فقط بينما كل الأوامر المتعلقة ب**sql** (أي جميع الاستعلامات واوامر الإضافة وغيرها) كلها سوف توضع على مستوى السيرفر.

ان الفكرة الأساسية للإجراءات المخزنة موحدة في جميع قواعد البيانات لكن الطريقة التي تطبق فيها قواعد البيانات الإجراءات المخزنة تختلف, ولذلك فان استدعاء الإجراءات المخزنة يتم بواسطة تعبيرات تعتمد على قاعدة البيانات كما يلي :

**\* في قاعدة البيانات Oracle و DB2**

يستخدم التعبير **CALL** .

**\* في قواعد البيانات SQL Server و Access**

يستخدم التعبير **EXECUTE** او **EXEC** .

## مثال على الإجراءات المخزنة:



في البداية نحتاج الى برنامج xampp مثلا فندخل على sql

ونقوم بكتابة الكود الاتي داخل sql لإنشاء قاعدة البيانات باسم **employee**

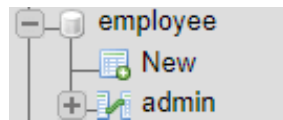
```
Create database `employee` ;
```

بعد كتابة الكود نضغط على زر go في الأسفل من جهة اليمين

ثم نقوم بإنشاء جدول باسم **admin** داخل قاعدة **employee** في sql

```
Create table `admin`
```

```
( id int primary key , name varchar (50) , age int (50) , email varchar (50) );
```



بعد الانتهاء من كتابة هذا الكود نضغط على زر go

قمنا بإنشاء قاعدة بيانات وجدول كما موضح بالصورة

وبعدها نقوم بإدخال بيانات معينة الى جدول **admin** في sql

```
Insert into `admin` ( id , name , age , email)
```

```
Values( 1 , 'Rahma' , 22 , 'rahma @ gmail.com' );
```

```
Insert into `admin` ( id , name , age , email)
```

```
Values( 2 , 'Omer' , 25 , 'omer@gmail.com' );
```

```
Insert into `admin` ( id , name , age , email)
```

```
Values( 3 , 'Ali' , 28 , 'ali@gmail.com' );
```

بعد الانتهاء من كتابة الكود نضغط على زر go

فيظهر لنا هذا الجدول في قاعدة البيانات ويحتوي على جميع البيانات التي قمنا بأدخالها

Id	Name	Age	Email
1	Rahma	22	rahma @ gmail.com
2	Omer	25	omer @ gmail.com
3	Ali	28	ali @ gmail.com

والان بعد ان قمنا بتكون جدول admin نستطيع عمل stored procedure عليه

مثال: إنشاء اجراء مخزن يقوم بأستدعاء جميع البيانات:

```
DELIMITER //  
CREATE PROCEDURE all_emp ()  
BEGIN  
Select * From admin ;  
END //  
DELIMITER ;
```

في السطر الأول والأخير قمنا بأستخدام الامر delimiter وهو ليس جزءا من الاجراء المخزن

ففي السطر الأول: يقوم هذا الامر بتغيير المحدد الافتراضي الى //

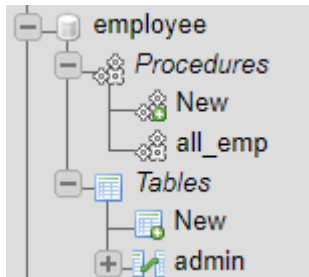
وفي السطر الأخير: يقوم هذا الامر بتغيير المحدد مرة أخرى الى الافتراضي وهو الفاصلة المنقوطة ;

في السطر الثاني: قمنا بإنشاء stored procedure او اجراء مخزن في قاعدة البيانات وقمنا باعطائه اسم all\_emp لكي يتم حفظه واستدعائه بنفس الاسم.

في السطر الثالث والخامس: كتبنا الامر begin و end لوضع الاوامر داخلهم.

اما في السطر الرابع: قمنا بكتابة امر لاستدعاء جميع البيانات.

بعد كتابة هذا الكود نضغط على زر go ثم نعمل Refresh فتظهر لنا دالة الاجراء stored procedure وداخلها جدول all\_emp كما موضح في الصورة



بعد القيام بإنشاء هذا الاجراء الان نقوم باستدعائه لكي يتم تنفيذ الأوامر داخل البرنامج وبإحدى الطريقتين:

### الطريقة الأولى:

نقوم بكتابة الكود الاتي داخل sql

```
Call all_emp ();
```

وبعد الانتهاء من كتابة الكود نضغط على زر go

**ملاحظة:** نستخدم كلمة call لكي نستدعي الاجراء داخل procedure ثم نكتب بعده اسم الاجراء.

سوف تقوم هذه الدالة بأستدعاء جميع البيانات من جدول admin



**الطريقة الثانية:** نذهب الى الاجراء الذي قمنا بإنشاءه داخل procedure ونضغط عليه هنا فتظهر لنا هذه النافذة ونضغط على زر go فتتنفذ الدالة وتظهر لنا جميع البيانات

Execute routine `all\_emp`

Routine parameters

Name	Type	Function	Value
------	------	----------	-------

Go Close

**ملاحظة:** اذا قمنا بالضغط على اسم الاجراء في stored procedure سوف يظهر لنا هذا الكود الذي قمنا بكتابة في sql لكي نقوم بالتعديل عليه من هنا دون الذهاب الى sql مرة اخرى

all\_emp |

PROCEDURE

Direction	Name	Type	Length/Values	Options
-----------	------	------	---------------	---------

Add parameter

```
1 BEGIN
2 Select * From admin ;
3 END
```

مثال: أنشاء اجراء مخزن يقوم باستدعاء بيانات موظف واحد من جدول admin .

```
DELIMITER //
CREATE PROCEDURE one_emp ( in emp_user int )
BEGIN
Select * From admin where id = emp_user;
END //
DELIMITER ;
```

نستخدم الأمر in لاستقبال براميتر معين ونكتب بعدها اسم الحقل الجديد ونوعه

بعد الانتهاء من كتابة الكود نضغط على زر go

والان نقوم بعمل Refresh في البرنامج وسوف يظهر لنا الاجراء الذي قمنا بأنشاءه بأسم one\_emp داخل procedure  
بعد القيام بأنشاء هذا الاجراء الان نقوم باستدعاءه لكي يتم تنفيذ الأوامر داخل البرنامج وبإحدى الطريقتين:

**الطريقة الأولى:**

نقوم بكتابة الكود الاتي داخل sql

```
CALL one_emp (1);
```

وبعد الانتهاء من كتابة الكود نضغط على زر go

**الطريقة الثانية:** نضغط على اسم الاجراء كما ذكرنا سابقا فتظهر لنا هذه النافذه ونكتب فيها رقم الموظف الذي نريد استدعاء بياناته مثلا الموظف رقم واحد.

Name	Type	Function	Value
emp_user	INT		1

تم أستدعاء الاجراء وجلب الموظف رقم واحد مثلا :

Id	Name	Age	Email
1	Rahma	22	rahma @ gmail.com

مثال : أنشاء اجراء مخزن يقوم بأضافة حقل معين الى جدول admin مع استخدام الامر IN  
لاضافة حقول معينة وتكون هذه الحقول مساوية للحقول السابقة .

```
DELIMITER //
CREATE PROCEDURE add_emp (
in p_id int,
in p_name varchar (100) ,
in p_age int (100) ,
in p_email varchar (100)
)
BEGIN
insert into admin
values (p_id , p_name , p_age , p_email );
END //
DELIMITER;
```

نستخدم الأمر in لاستقبال براميتر واطافة بيانات الى الجدول  
ونقوم بكتابة أسماء حقول لجدول edit\_emp على شرط ان تكون نفس اسماء الحقول  
الموجودة في admin مع تغيير بسيط بالاسم  
وبعد الانتهاء من كتابة الاكواد نضغط الزر go

سوف يقوم بانشاء جدول بأسم add\_emp ويكون هذا الجدول داخل procedure  
بعد القيام بأنشاء هذا الجدول الان نقوم باستدعائه لكي يتم تنفيذ الأوامر داخل البرنامج وبإحدى  
الطريقتين:

## الطريقة الأولى:

نقوم بكتابة الكود الاتي داخل sql

```
CALL add_emp ( 4, 'ahmed' ,30 , 'ahmed@gmail.com' );
```

Id	Name	Age	Email
1	Rahma	22	rahma @ gmail.com
2	Omer	25	omer @ gmail.com
3	Ali	28	ali @ gmail.com
4	ahmed	30	ahmed@gmail.com

قام بإضافة الموظف احمد الى الجدول

والطريقة الثانية كما ذكرنا سابقا

مثال: إنشاء اجراء مخزن يقوم بتحديث جدول admin وأيضا استخدم الامر IN .

```
CREATE PROCEDURE edit_emp (  
in p_id int,  
in p_name varchar (100),  
in p_age int (100),  
in p_email varchar (100)  
)  
BEGIN  
UPDATE admin SET  
name = p_name,  
age = p_age,  
email = p_email  
WHERE name = p_name;  
END //  
DELIMITER ;
```



بعد الانتهاء من كتابة الكود نضغط على زر go سوف يقوم بإنشاء اجراء باسم edit\_emp ويكون هذا الاجراء داخل procedure

بعد القيام بإنشاء هذا الاجراء الان نقوم باستدعاءه لكي يتم تنفيذ الأوامر داخل البرنامج  
نقوم بكتابة الكود الاتي داخل sql

```
CALL edit_emp(3, 'Ali', 35, 'ali altaee@gmail.com' );
```

سوف يقوم بأستدعاء الاجراء وتحديث هذا الحقل في جدول admin

Id	Name	Age	Email
1	Rahma	22	rahma @ gmail.com
2	Omer	25	omer @ gmail.com
3	Ali	35	ali altaee @ gmail.com
4	ahmed	30	ahmed@gmail.com

عند الدخول الى جدول admin سوف نرى انه قام بتحديث بيانات الموظف رقم 3 لكنه لم يتم بتحديث اسمه كما في الشرط الذي كتبناه سابقا

مثال: قم بحذف بيانات في جدول admin بواسطة الإجراءات المخزنة.

```
DELIMITER //
```

```
CREATE PROCEDURE remove_emp (in p_age int )
```

```
BEGIN
```

```
DELETE FROM admin WHERE age=p_age ;
```

```
END //
```

```
DELIMITER ;
```

قمنا بكتابة امر لحذف بيانات الموظف عن طريق العمر

بعد الانتهاء من كتابة الكود نضغط على زر go سوف يقوم بإنشاء جدول باسم remove\_emp ويكون هذا الجدول داخل procedure

بعد القيام بإنشاء هذا الجدول الان نقوم باستدعاءه لكي يتم تنفيذ الأوامر داخل البرنامج  
نقوم بكتابة الكود الاتي داخل sql

```
call remove_emp(25);
```

Id	Name	Age	Email
1	Rahma	22	rahma @ gmail.com
3	Ali	35	ali altaee @ gmail.com
4	ahmed	30	ahmed@gmail.com

عند الدخول الى جدول admin سوف نرى انه قام بحذف الموظف الثاني الذي كان عمره 25 سنة

\*\*\*\*\*

سوف نشرح بشكل بسيط عن **variables** (المتغيرات) وكيفية التعامل معهم في **Mysql** من حيث الجمع وإظهار قيمة لمتغير ما:

مثال : اظهار قيمة للمتغير الأول تساوي 5 والمتغير الثاني كلمة hello

في برنامج xampp داخل sql نقوم بكتابة الكود الاتي

```
SET @variable1 = 5;
```

```
SELECT @variable2 := 'hello' ;
```

+ Options

```
@variable2 := 'hello'
```

```
hello
```

وبعدها نضغط على زر go فنظهر النتيجة كما موضح بالصورة

مثال : أظهار المتغيرات الاتية 1,2,3 مع إعطاء قيمة لكل متغير.

في برنامج xampp داخل sql نقوم بكتابة الكود الاتي

```
SET @variable1 = 5;
```

```
SET @variable2 = 10;
```

```
SET @variable3 = -7;
```

```
SELECT @variable1, @variable2, @variable3;
```

+ Options

```
@variable1 @variable2 @variable3  
5 10 -7
```

وبعدها نضغط على زر go فتظهر النتيجة كما موضح  
بالصورة

مثال : أظهار المتغيرات الاتية 1,2,3 مع إعطاء قيمة لكل متغير وايضا قم بجمعهم.

```
SET @variable1 = 5;
```

```
SET @variable2 = 10;
```

```
SET @variable3 = -7;
```

```
SELECT @variable1, @variable2, @variable3,
```

```
@variable1+ @variable2+ @variable3;
```

وبعدها نضغط على زر go فتظهر النتيجة كما موضح بالصورة

+ Options

```
@variable1 @variable2 @variable3 @variable1+@variable2+@variable3  
5 10 -7 8
```

مثال : أظهار المتغيرات الاتية 1,2,3 مع إعطاء قيمة لكل متغير وايضا قم بجمعهم وضع  
النتيجة في المتغير 4.

```
set @variable1=4;
```

```
SET @variable2 = 10;
```

```
SET @variable3 = -7;
```

```
SELECT @variable1, @variable2, @variable3, @variable4:=
```

```
@variable1+ @variable2+ @variable3;
```

```
SELECT @variable4;
```

وبعدها نضغط على زر go فتظهر النتيجة كما موضح بالصورة

+ Options

```
@variable1 @variable2 @variable3 @variable4:= @variable1 + @variable2 + @variable3  
4 10 -7 7
```

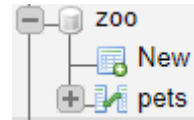
+ Options

```
@variable4  
7
```

سوف نقوم بعمل قاعدة بيانات جديدة باسم zoo وجدول باسم pets في برنامج xampp داخل  
sql

```
CREATE DATABASE `zoo` ;  
CREATE TABLE `pets`  
(id int PRIMARY KEY AUTO_INCREMENT , name varchar(50) , owner  
varchar(50), species varchar(50), sex char(1), birth date , death date);
```

وبعدها نضغط على زر go فتظهر لنا قاعدة بيانات zoo وجدول pets



كما موضح بالصورة

وبعدها نقوم بإدخال بيانات معينة الى جدول pets في sql

```
INSERT INTO pets  
(id, name, owner, species, sex, birth, death)  
VALUES(1, 'Snowflake', 'Ewin', 'Robo Hamster', 'F', '2017-03-13', NULL) ;
```

```
INSERT INTO pets  
(id, name, owner, species, sex, birth, death)  
VALUES(2, 'Cuddles', 'Frank', 'kittly', 'M', '2018-11-11', '2018-12-01) ;
```

```
INSERT INTO pets  
(id, name, owner, species, sex, birth, death)  
VALUES(3, 'Princess', 'Sam', 'Horsey', 'F', Now(), NULL) ;
```

وبعدها نضغط على زر go فيظهر لنا جدول pets في قاعدة البيانات ويحتوي على جميع البيانات التي قمنا بأدخالها

Id	Name	Owner	Species	Sex	Birth	Death
1	Snowflake	Ewin	Robo Hamster	F	2017-03-13	NULL
2	Cuddles	Frank	Kittly	M	2018-11-11	2019-12-01
3	Princess	Sam	Horsey	F	Now()	NULL

لنأخذ امثلة على المتغيرات وكيفية التعامل معها داخل الجداول

مثال : قم بأظهار العنصر الأول(id1) من جدول pets الموجود في قاعدة البيانات zoo داخل mysql

```
SET @id = 1;
SELECT * FROM pets
WHERE id=@id;
```

ثم نضغط على زر go فتظهر النتيجة كالآتي

id	name	owner	species	sex	birth	death
1	Snowflake	Ewin	Robo Hamster	F	2017-03-13	NULL

مثال : قم بأظهار العنصر الأول (id1) من جدول pets الموجود في قاعدة البيانات zoo داخل mysql وتكون النتيجة في (id2)

```
SET @id = 1;
SELECT @id :=2;
SELECT * FROM pets
WHERE id=@id;
```

+ Options

@id :=2

2

نضغط على زر go بعد كتابة الكود فتظهر النتيجة كالآتي

id	name	owner	species	sex	birth	death
2	Cuddles	Frank	kittly	M	2018-11-11	2019-12-01

سؤال :. اذا قمنا بتغيير SELECT @id :=2 الى SELECT@id =2 ماذا يحصل؟؟ لنرى في المثال الاتي:

```
SET @id = 1;
SELECT@id =2;
SELECT * FROM pets
WHERE id=@id;
```

بعد الانتهاء نضغط على زر go وكما موضح بالصورة

+ Options  
@id =2  
0

id	name	owner	species	sex	birth	death
1	Snowflake	Ewin	Robo Hamster	F	2017-03-13	NULL

مثال اخر على السؤال السابق لكن id=5:

```
SET @id = 5;
SELECT @id =5;
SELECT * FROM pets
WHERE id=@id;
```

وتظهر النتيجة كالآتي

+ Options  
@id =5  
1

id	name	owner	species	sex	bith	death
----	------	-------	---------	-----	------	-------

فتظهر لنا النتيجة فارغة لاننا لم ندم بأدخال بيانات في id 5

نقوم بعمل الإجراءات المخزنة **stored procedure** على قاعدة البيانات والجدول السابق  
(zoo, pets) وكالاتي:

مثال : أنشاء اجراء مخزن واسترجاع البيانات الذي يكون فيها Sex من نوع F فقط من  
الجدول pets

```
DELIMITER //  
CREATE PROCEDURE pets_female ()  
BEGIN  
SELECT * FROM pets  
WHERE sex = "F";  
END //  
DELIMITER ;
```

بعد الانتهاء من كتابة الكود نضغط على زر go  
بعد ذلك نقوم بأستدعاء الاجراء او الجدول فنكتب الكود الاتي:

```
CALL pets_female();
```

فيظهر لنا هذا الجدول

Id	Name	Owner	Species	Sex	Birth	Death
1	Snowflake	Ewin	Robo Hamster	F	2017-03-13	NULL
3	Princess	Sam	Horsey	F	2020-03-01	NULL

مثال اخر: أنشاء اجراء مخزن واسترجاع البيانات التي تكون فيها sex من نوع M و يكون  
التاريخ birth هو 2018-11-11

```
DELIMITER //  
CREATE PROCEDURE pets_female_with_birth ()  
BEGIN  
SET @birthdate := "2018-11-11";
```

```

SELECT * FROM pets
WHERE sex = "M"
AND birth = @birthdate ;
END //
DELIMITER ;

```

ثم نضغط على زر go

بعد ذلك نقوم بأستدعاء الاجراء او الجدول كالاتي:

```
CALL pets_female_with_birth();
```

فيظهر لنا هذا العنصر فقط

Id	Name	Owner	Species	Sex	Birth	Death
2	Cuddles	Frank	Kittly	M	2018-11-11	2019-12-01

\*\*\*\*\*

هنالك ثلاثة أنواع من العوامل في الMysql

**IN :** وهو القيمة الافتراضية لكل عامل عند تصريح العامل IN تستطيع من خلالها جلب قيمة من خارج الوحدة وتستطيع تغيير قيمة هذا العامل داخل الوحدة ولكن بدون ارسالها خارج الوحدة.

**OUT :** قيمة تبدء بnull وتستطيع تغيير قيمتها ثم ارسال هذه القيمة الجديدة خارج الوحدة .

**INOUT :** وهي عبارة عن دمج النوعان السابقان.



لنقوم بعمل قاعدة بيانات جديدة باسم World وجدول باسم City في برنامج xampp داخل sql

ونقوم بكتابة الكود الاتي داخل sql لإنشاء قاعدة البيانات باسم world وجدول باسم city

```
CREATE DATABASE `world`;  
  
CREATE TABLE `city`  
  
( Id int PRIMARY KEY , Name varchar(25) , Countrycode varchar (25) ,  
District varchar(25) , Population int (25)  
  
);
```

وبعدھا نقوم بإدخال بيانات معينة الى جدول city في sql

```
INSERT INTO `city`  
(`Id`,`Name`,`Countrycode`,`District`,`Population`)  
VALUES (1,'Kabul' , 'AFG', 'Kabol' ,178000);  
  
INSERT INTO `city`  
(`Id`,`Name`,`Countrycode`,`District`,`Population`)  
VALUES (2,'Tafuna' , 'ASM', 'Tutuila' ,5200);  
  
INSERT INTO `city`  
(`Id`,`Name`,`Countrycode`,`District`,`Population`)  
VALUES (3,'Fagatogo' , 'ASM', 'Tutuila' ,23230);  
  
INSERT INTO `city`  
(`Id`,`Name`,`Countrycode`,`District`,`Population`)  
VALUES (4,'Luanda' , 'AGO', 'Luanda' ,202200);
```

بعد الانتهاء نضغط على زر go فيظهر لنا الجدول الاتي وفيه البيانات التي قمنا بإدخالها

Id	Name	Countrycode	District	Population
1	Kabul	AFG	Kabol	178000
2	Tafuna	ASM	Tutuila	5200
3	Fagatogo	ASM	Tutuila	23230
4	Luanda	AGO	Luanda	202200

الآن نبدأ بعمل stored procedure لجدول city الذي قمنا بعمله

مثال: إنشاء اجراء مخزن واسترجاع جميع البيانات من جدول city .

```
DELIMITER $$  
CREATE PROCEDURE get_all_cities()  
BEGIN  
SELECT * FROM city ;  
END $$  
DELIMITER ;
```

**ملاحظة:** نستطيع ان نستخدم الرمز \$\$ او أي رمز بدلا من // في السطر الأول و قبل الأخير من الكود

لكن بشرط :. ان يكون نفس الرمز المستخدم في البداية والنهاية

الآن نقوم بأستدعاء الاجراء وكالاتي

```
CALL get_all_cities();
```

مثال اخر: إنشاء اجراء مخزن واسترجاع جميع البيانات وادخال قيمة عن طريق IN لاضافة حقل معين ويكون هذا الحقل الجديد مساوي لحقل id السابق .

```
DELIMITER $$  
CREATE PROCEDURE get_city ( in City_Id INT )  
BEGIN  
SELECT * FROM city  
WHERE Id = City_Id ;  
END $$  
DELIMITER ;
```

نستخدم الأمر in لاستقبال او جلب قيمة معينة ونكتب بعدها اسم الحقل الجديد ونوعه

بعد الانتهاء من كتابة الكود نضغط على زر go

**ملاحظة مهمة:** توجد طريقتان لاستدعاء الاجراء الموجود في stored procedure وهي كما ذكرنا سابقا:

**الطريقة الأولى:** الضغط على اسم الاجراء هنا  ثم تظهر لنا نافذة ونضغط على go .

**اما الطريقة الثانية:** وهي كتابة الامر call ثم اسم الجدول او الاجراء كما شرحنا سابقا بطريقة واضحة ومبسطة.

**والان توجد طريقة ثالثة:** لاستدعاء الاجراء وهي كتابة الامر call ثم اسم قاعدة البيانات التابعة لهذا الجدول ثم **(نقطة)** ثم اسم الاجراء المراد استدعاءه وكما موضح بالكود الاتي.

```
call world . get_city (2);
```

سوف يظهر لنا جدول وفيه بيانات العنصر الثاني :

Id	Name	Countrycode	District	Population
2	Tafuna	ASM	Tutuila	5200

**مثال :** انشاء اجراء مخزن لأسترجاع Name و District فقط من جدول city وادخال قيمة عن طريق IN .

```
DELIMITER $$
```

```
CREATE PROCEDURE getCityInfo ( in Population_input int (25) ,
```

```
in Countrycode_input varchar (25))
```

```
BEGIN
```

```
SELECT Name , District FROM city
```

```
WHERE Population = Population_input
```

```
AND Countrycode = Countrycode_input ;
```

```
END $$
```

```
DELIMITER ;
```

```
اما CALL getCityInfo(23230,"ASM");
```

```
او CALL world . getCityInfo(23230,"ASM");
```

من الجدير بالذكر انه يمكننا انشاء الاجراء واستدعاءه بنفس الوقت وبنفس لوحة sql

بعد الانتهاء من كتابة كود الانشاء والاستدعاء نضغط على زر go فيظهر لنا هذا الجدول

Name	District
Fagatogo	Tutuila

+ Options  
Name District  
Fagatogo Tutuila

مثال : انشاء اجراء مخزن وادخال قيمة عن طريق IN لحقل id وإخراج قيمة عن طريق OUT لحقل name من جدول city

DELIMITER \$\$

CREATE PROCEDURE getCityName

( in City\_Id int , out City\_Name varchar (25))

BEGIN

SELECT Name INTO City\_Name

FROM city

WHERE Id = City\_Id ;

END \$\$

DELIMITER ;

نستخدم الأمر in لاستقبال او جلب قيمة معينة ونكتب بعدها اسم الحقل الجديد ونوعه.

والامر out لاجراج او عرض قيمة معينة وأيضا نكتب بعدها اسم الحقل الجديد ونوعه

بعد الانتهاء من كتابة الكود نضغط على زر go

ثم نكتب بعده امر لاستدعاء الاجراء وكالاتي:

SET @City\_Id =1 ;

CALL getCityName(@City\_Id,@City\_Name);

SELECT @City\_Name;

في السطر الأول قمنا بعمل set لحقل id أي انه يمكننا استرجاع أي عنصر بمجرد كتابة id الخاص به من جدول city فيقوم بأسترجاع حقل name كما موضح بالصورة:

+ Options  
@City\_Name  
Kabul

مثال : قم بكتابة كود لحذف وانشاء اجراء مخزن من جدول city في قاعدة البيانات world ثم قم بعد ذلك بأسترجاع جميع البيانات الموجودة في جدول city

```
USE world ;
DROP PROCEDURE IF EXISTS getcities ;
DELIMITER $$
CREATE PROCEDURE getcity()
BEGIN
SELECT * FROM city ;
END $$
DELIMITER ;
CALL getcity();
```

في البداية قمنا بكتابة قاعدة البيانات الخاصة بجدول city  
ثم قمنا بكتابة كود لحذف الاجراء المخزن الذي قمنا بعمله سابقا وبعد ذلك قمنا بكتابة كود لانشاء اجراء مخزن جديد

بعد كتابة كود الانشاء والاستدعاء نضغط على زر go

فيظهر لنا هذا الجدول:

Id	Name	Countrycode	District	Population
1	Kabul	AFG	Kabol	178000
2	Tafuna	ASM	Tutuila	5200
3	Fagatogo	ASM	Tutuila	23230
4	Luanda	AGO	Luanda	202200

مثال : انشاء اجراء مخزن لعرض أي عنصر في جدول city بمجرد كتابة Countrycode الخاص به وادخال قيمة عن طريق IN.

```
DELIMITER $$
CREATE PROCEDURE getCitiesBasedCountryCode (in Countrycode_input
varchar (35) )
BEGIN
```

```

SELECT * FROM city
WHERE Countrycode = Countrycode_input ;
END $$
DELIMITER ;
CALL getCitiesBasedCountryCode ("AFG");
CALL getCitiesBasedCountryCode ("AGO");

```

بعد الانتهاء من كتابة الكود اضغط على زر go فيظهر لنا الجدول الاتي:

<b>Id</b>	<b>Name</b>	<b>Countrycode</b>	<b>District</b>	<b>Population</b>
1	Kabul	AFG	Kabol	178000
4	Luanda	AGO	Luanda	202200

مثال : انشاء اجراء مخزن وادخال قيمة عن طريق IN وإخراج قيمة عن طريق OUT لاستقبال وعرض جميع البيانات من جدول city

```

DELIMITER $$
CREATE PROCEDURE getinfo1(
in city_Id_input int ,
in city_Population_input int ,
out city_Name_output varchar (50) ,
out city_Countrycode_output varchar (50) ,
out city_District_output varchar (50) ,
out city_Population_output int (50)
)
BEGIN
SELECT Name , Countrycode , District , Population INTO
city_Name_output ,

```

```

city_Countrycode_output ,
city_District_output ,
city_Population_output
FROM city
WHERE Id = city_Id_input AND Population > city_Population_input ;
END $$
DELIMITER ;
SET @city_Id_input = 3;
SET @city_Population_input =1000 ;
CALL getinfo1(
@city_Id_input ,
@city_Population_input ,
@city_Name_output ,
@city_Countrycode_output ,
@city_District_output ,
@city_Population_output );
SELECT
@city_Name_output ,
@city_Countrycode_output ,
@city_District_output ,
@city_Population_output;

```

بعد الانتهاء من كتابة الكود نضغط على زر go  
ثم تظهر لنا بعد ذلك هذه النتيجة كما موضح بالصورة:

+ Options	@city_Name_output	@city_Countrycode_output	@city_District_output	@city_Population_output
	Fagatogo	ASM	Tutuila	23230

فتظهر لنا جميع الحقول ولكن بالأسماء التي قمنا بأدخالها عن طريق الامر OUT

تَعْمِدُ بِحَمْدِ اللَّهِ