جامعة طرابلس
كلية تقنية المعلومات

Advanced Databases قواعد البيانات المتقدمة
ITSE312

أستاذ المادة – حسن علي حسن

h.ebrahem@uot.edu.ly

المحاضرة الثامنة – المعاملة والإجراء المخزن

Transaction, Stored Procedure

مواضيع المحاضرة الثامنة

- إدارة معالجة المعاملات Managing Transaction Processing
- الاجراءات المخزنة STORED PROCEDURES
  - مزايا الاجراء المخزن Advantages STORED PROCEDURE
  - عيوب الاجراء المخزن Disadvantages STORED PROCEDURE
  - إنشاء الاجراء المخزن CREATE STORED PROCEDURE
  - تصريح عن المتغير DECLARE OF Variables
  - معامل الإدخال IN Parameter
  - معامل الإخراج OUT Parameters
  - معامل الإدخال والإخراج INOUT Parameter
- الجملة الشرطية IF Statement

Managing Transaction Processing إدارة معالجة المعاملات

- **المعاملات Transactions** هي مجموعة من تعليمات SQL التي تنفذ بترتيب منطقي على جداول قاعدة البيانات، سواء كان التنفيذ عن طريق المستخدم أو تلقائيًا بواسطة برنامج قاعدة البيانات والتي قد تتم بشكل غير صحيح وتسبب في حدوث خطأ.
- بفرض أنه تم تنفيذ مجموعة من عمليات التحديث التي قامت بها SQL من (إدخال وتعديل وحذف) على عدة جداول في قاعدة البيانات، وأثناء التنفيذ حدث خطأ على سبيل المثال، نتيجة خطأ في الشبكة، نفاد مساحة التخزين، انقطاع الكهرباء ... إلخ). للمحافظة على سلامة البيانات من هذه الأخطاء التي قد تحدث، نقوم بترجيع البيانات إلى وضعها التي كانت عليه قبل القيام بتنفيذ عمليات التحديث، يتم ذلك عن طريق **معالجة المعاملات Transaction Processing**.

Managing Transaction Processing إدارة معالجة المعاملات

- في معالجة المعاملات Transaction Processing لا يتم تخزين البيانات فعليا داخل الجداول نتيجة تنفيذ عمليات التحديث إلا بعد تنفيذ أمر الاعتماد **COMMIT**، أما بالنسبة لأمر التراجع **ROLLBACK** فيتم التراجع عن عمليات التحديث التي تمت على بيانات الجداول، أي لا يتم تخزينها فعليا وتبقى الجداول على الحالة التي كانت عليها قبل تنفيذ معالجة المعاملات.

أمر الاعتماد COMMIT وأمر التراجع ROLLBACK

- عند القيام بمعالجة المعاملات Transaction Processing يتم اعتماد التحديثات التي تمت على البيانات التي في الجداول عن طريق أمر الاعتماد Commitأو التراجع عن التحديثات التي تمت عن طريق أمر التراجع Rollback.
- توجد بعض **التعليمات التي لا يقوم أمر ROLLBACK بمعالجتها**، بمعنى التراجع إلى ما كانت عليه الجداول قبل تنفيذ العمليات، هذه التعليمات في الشكل.

CREATE / ALTER / DROP DATABASE	**إنشاء / تعديل / حذف قاعدة البيانات**
CREATE /ALTER / DROP TABLE	**إنشاء / تعديل / حذف جدول**
CREATE / DROP INDEX	**إنشاء / تعديل / حذف فهرس**
CREATE / DROP PROCEDURE	**إنشاء / تعديل / حذف إجراء مخزن**

أمر الاعتماد COMMIT وأمر التراجع ROLLBACK

- بالنسبة إلى **التعليمات التي يقوم أمر COMMIT و ROLLBACK بمعالجتها**، أي يتم اعتماد تخزين البيانات فعليا داخل الجداول أو التراجع عن تخزينها، هذه التعليمات في الشكل التالي:

INSERT, UPDATE, DELETE	**إدخال، تعديل، حذف**

- أمر الاعتماد **COMMIT** يقوم باعتماد تخزين نتائج جميع العمليات الموجودة في الشكل، أي التي تم تنفيذها على سجلات الجداول من حذف أو تعديل أو إدخال.
- أمر التراجع **ROLLBACK**يقوم بترجيع نتائج كل العمليات في الشكل، أي يقوم بعدم تخزين بيانات أي عملية تم تنفيذها من حذف أو تعديل أو إدخال، سواء تم تنفيذ هذه العمليات بعد تنفيذ أمر COMMIT أو أمر ROLLBACK.

أمر الاعتماد COMMIT

- تختلف أنظمة إدارة قواعد البيانات DBMS في طريقة بدء التعامل مع المعاملة Transaction.
- في نظام إدارة قواعد البيانات MYSQL يجب تعطيل أمر COMMIT قبل البدء في معالجة المعاملات لكي لا يتم تخزين التحديثات التي تمت على الجداول حتى إنهاء كافة العمليات المطلوبة. يتم تعطيل أمر Commit باستخدام أمر AUTOCOMMIT، كما في الشكل

```
SET autocommit = 0;
```

- في حالة إعادة **تفعيل** هذا الأمر COMMIT لكي يشتغل بشكل آلي، يتم تغيير قيمة الأمر من 0 إلى القيمة **1**.

أمر الاعتماد COMMIT

- **مثال 1**:يوضح كيف يتم القيام بإدارة معاملة تحتوي مجموعة من عمليات التحديث باستخدام أمر Commit، كما في الشكل

```
SET autocommit = 0;
START TRANSACTION
DELETE  الزبون FROM  الزبون T1 LEFT join  التأجيرT2
USING( رقم_الزبون) WHERE T2.رقم_الملكية  IS NULL ;
UPDATE  الملكية SET الإيجار_الشهري = الإيجار_الشهري * 0.1 ;
INSERT INTO  مالك_العقار SELECT  *FROM  الزبون AS S1
WHERE EXISTS (SELECT * FROM  التأجير S2 WHERE
S1.رقم_الزبون = S2.رقم_الزبون);
COMMIT;
SET autocommit = 1;
```

أمر التراجع ROLLBACK

- **مثال 2**:البدء في إدارة للمعاملات لمجموعة من العمليات باستخدام أمر Rollback، كما في الشكل

```
SET autocommit = 0;
BEGIN TRANSACTION
UPDATE التأجير T1 inner join الملكيةT2 on T1.رقم_الملكية = T2. رقم_الملكية
SET T1.تاريخ_النهاية = "2020-12-31" WHERE T2.عنوان الملكية = "بن عاشور"
AND T1. تاريخ_النهاية IS NULL ;
DELETE FROM التأجير;
ROLLBACK;
SET autocommit = 1;
```

أمر التراجع ROLLBACK

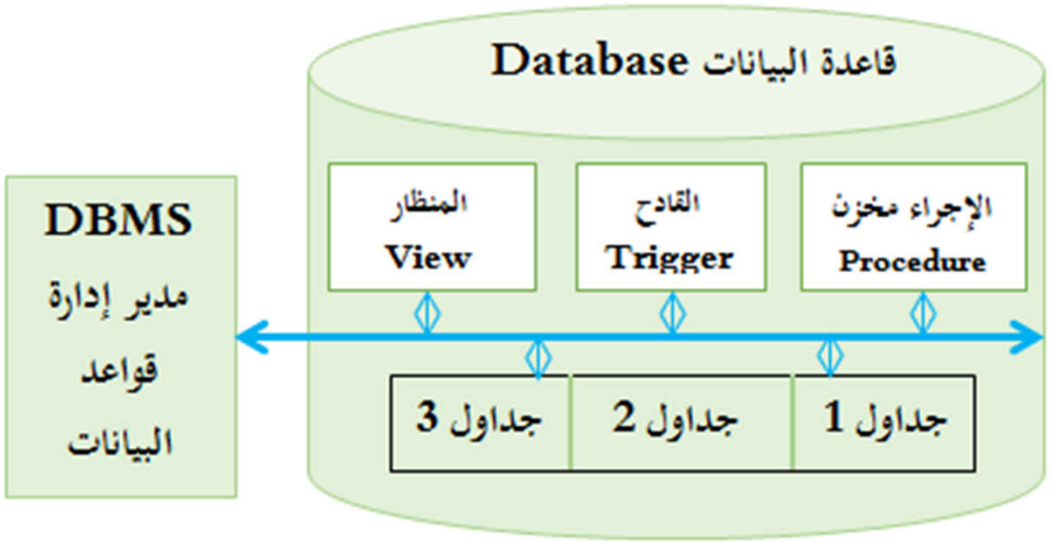
- نلاحظ، تم إجراء العديد من العمليات على جداول قواعد البيانات، تم تنفيذ عملية تعديل وحذف للبيانات على جدولين. بفرض حدوث خطأ داخل النظام قبل الانتهاء من جميع العمليات، يتم تنفيذ أمر ROLLBACK، حيث يقوم DBMS بإلغاء جميع العمليات السابقة داخل TRANSACTION التي تمت على الجدول من تعديل وحذف، وذلك بترجيع جدول التأجير إلى وضعه الطبيعي قبل عملية التعديل والحذف، ويبقى الجدول كما كان عليه قبل تنفيذ المعاملة TRANSACTION، ثم يتم إرجاع النظام إلى الوضع الافتراضي باستخدام SET AUTOCOMMIT = 1 .

الإجراء المخزن Stored Procedure

- جملة الاستفسار (الاستعلام) Select تقوم باسترجاع مجموعة من السجلات من جداول مختلفة، وفي بعض الأحيان قد يتم كتابة جملة استفسار متداخلة من أكثر من جملة عند التعامل مع السجلات داخل الجداول أو قد نحتاج إلى إجراء بعض التحديثات داخل الجداول (إدخال، حذف، تعديل). بدل من كتابة هذه الجملة عند التعامل مع الجداول، يمكن أن يتم تخزين هذه الجملة داخل قاعدة البيانات ويتم استدعائه عند الحاجة إليها بدون كتابتها من جديد، تتم عملية التخزين هذه داخل الإجراء المخزن Stored Procedure.

الإجراء المخزن Stored Procedure

- الإجراء المخزن Stored Procedure، يعتبر الإجراء من الميزات أكثر تقدما في لغة SQL والتي تخص قواعد البيانات العلائقية، حيث يتم تخزينه مع الجداول المرتبط بها في قاعدة البيانات ويتركب الاجراء داخليا من جملة الاستفسار SELECT. أنظر الشكل.



الإجراء المخزن Stored Procedure

- الاجراء المخزنة هو عبارة عن مقطع من تعليمات لغة الاستفسار SQL التي يتم تخزينها مع الجداول في قاعدة البيانات. يمكن استدعاء الاجراء المخزن عن طريق إجراء مخزن أخر أو قادح TRIGGER أو أي برنامج Program.
- بعض أنظمة إدارة قواعد البيانات DBMS لا تدعم الإجراء المخزن مثل Microsoft Access، ينصح بالاطلاع على وثائق نظام DBMS الخاصة به للحصول على مزيد من المعلومات.

Advantages STORED PROCEDURE مزايا الاجراء المخزن

1. **زيادة في كفاءة أداء التطبيقات Increase Efficiency**: يكون تنفيذ الاجراءات المخزنة عادة اسرع من تنفيذ تعليمات SQL التي ترسل عن طريق التطبيقات، وذلك لأن الإجراءات المخزنة تم ترجمتها مسبقا وتخزينها في قاعدة البيانات.

2. **التقليل من حجم تبادل البيانات Reducing of Data Exchange**: الاجراءات المخزنة تقلل من حجم تبادل البيانات بين التطبيقات وخادم قاعدة البيانات، لان التطبيق يقوم بإرسال أمر واحد لتنفيذ الاجراء المخزن بدلا من ارسال عدة أوامر عن طريق تعليمات SQL.

Advantages STORED PROCEDURE مزايا الاجراء المخزن

3. **إعادة الاستخدام Reuse**: يمكن إعادة استخدام الاجراءات المخزنة عن طريق تطبيقات مختلفة، وهذا يوفر على المبرمج إعادة كتابة تعليمات SQL في التطبيقات المختلفة.

4. **الاستخدام الآمن Safe**: يعتبر الإجراء المخزن آمن، حيث يمكن لمدير قاعدة البيانات أن يمنح إذن الوصول والاستخدام للإجراءات المخزنة للتطبيقات.

عيوب الاجراء المخزن Disadvantages STORED PROCEDURE

1. يحتوي الإجراء المخزن على تعليمات SQL فقط، ولا يمكن كتابة تعليمات مثل تلك التي تكتب في لغات البرمجة الاخرى.
2. صعوبة تتبع الأخطاء التي تنتج عن الاجراءات المخزنة في معظم أنظمة إدارة قواعد البيانات.
3. إذا تم إجراء تغيرات في بنية الجداول (مثل تغيير اسماء الخصائص)، يتطلب هذا التغيير تحديث في تركيبة الإجراء المخزن.

CREATE STORED PROCEDURE إنشاء الاجراء المخزن

تختلف تركيبة الإجراء المخزن على حسب نظام إدارة قواعد البيانات DBMS، الشكل يبين تركيبة الإجراء المخزن في نظام MYSQL وORACLE.

```
DELIMITER //
CREATE  PROCEDURE  اسم الإجراء( معامل)
  Begin
            جملة الاستفسار أو أوامر التحديث;
  End //
DELIMITER ;
```

CREATE STORED PROCEDURE إنشاء الاجراء المخزن

مثال: إنشاء إجراء مخزن باسم أول_زبون يقوم بعرض أول سجل من جدول الزبون. أنظر الشكل

```
DELIMITER //
CREATE  PROCEDURE   أول_زبون( )
   Begin
         SELECT * FROM  الزبون LIMIT 1 ;
   End //
DELIMITER ;
```

جدول الزبون

رقم_الزبون	اسم_الزبون
100	أحمد محمد
200	العلام عاصم
300	عبد المعز خيري
400	إبراهيم عبدالجواد

STORED PROCEDURE استدعاء الاجراء المخزن

بعد ما يتم إنشاء الاجراء المخزن داخل قاعدة البيانات، يتم استدعائه بالأمر في الشكل

نظام MYSQL	CALL اسم الاجراء();

- **تنبيه:** دائما يتم استخدام الأقواس مع اسم الاجراء المخزن، سواء أكان بداخلها قيم أو لا.

- عند استدعاء الإجراء عن طريق جملة الاستدعاء CALL يجب اتباع الإرشادات التالية:

1. ما بين الأقواس في جملة الاستدعاء CALL يجب أن يحتوي على نفس عدد القيم المحددة في الإجراء.
2. ترتيب القيم ما بين الأقواس في جملة الاستدعاء يكون بنفس ترتيب القيم المحددة في الإجراء.
3. نوع القيم ما بين الأقواس في جملة الاستدعاء يجب أن يتوافق مع نوع القيم في الاجراء.

STORED PROCEDURE استدعاء الاجراء المخزن

الاجراء المخزن (أول_زبون) الذي تم إنشائه، لا يعرض نتيجة إلا إذا تم استدعائه، لاستدعاء الاجراء المخزن، نستخدم الأمر كما في الشكل التالي:

```
CALL أول_زبون() ;
```

عند استدعاء الإجراء المخزن يقوم نظام DBMS بتنفيذ الإجراء داخل قاعدة البيانات، حيث يتم تنفذ الجمل التي بداخل الإجراء، نتيجة استدعاء الإجراء تظهر في الجدول.

رقم الزبون	اسم الزبون
100	أحمد محمد

المتغيرات Variables

يمكن استخدام المتغيرات Variables مع الاجراء المخزن، تستخدم المتغيرات في تخزين قيم بداخلها، المتغيرات يتم التصريح عليها قبل تنفيذ الاجراء أو داخل الاجراء المخزن.

```
DECLARE اسم المتغير نوع المتغير(حجم) DEFAULT القيمة المبدئية;
```

للتصريح عن مجموعة من المتغيرات Variables، أنظر الشكل:

```
DECLARE TOTAL INT DEFAULT 0 ;
DECLARE x, y INT DEFAULT 0 ;
```

DECLARE OF Variables تصريح عن المتغيرات

يمكن استخدام SELECT INTO داخل الإجراء المخزن لاسترجاع قيمة من جدول وتخصيص القيمة لمتغير، أنظر الشكل

```
DECLARE  TOTAL INT  DEFAULT  0 ;
SELECT COUNT(*) INTO TOTAL FROM مالك_العقار;
```

- تستخدم هذه الصيغة داخل الاجراء المخزن لاسترجاع سجلات من جدول وتخصيص القيمة في متغير.
- أما في حالة تخصيص قيمة لمتغير خارج الاجراء، نستخدم Set كما في الشكل

```
DECLARE  TOTAL INT  DEFAULT  0 ;
SET TOTAL = 100;
```

المعاملات Parameters

داخل الأقواس في الاجراء المخزن يمكن أن نستخدم المعاملات Parameters، تستخدم هذه المعاملات في ارسال قيمة أو متغير الى الاجراء المخزن لكي يتم الاستفادة منها داخل الاجراء، المعامل Parameter يجعل الإجراء أكثر مرونة وفاعلية. توجد ثلاثة أنواع من المعاملات التي تستخدم مع الإجراء: معامل إدخال IN، معامل إخراج OUT، معامل إدخال وإخراج INOUT.

يتم تعريف المعامل Parameter داخل الاقواس في الاجراء كما في الشكل

MODE اسم المعامل نوع المعامل(حجم المعامل)

معامل الإدخال IN Parameter

يستخدم معامل الإدخال IN Parameter مع الإجراء لتمرير قيمة لداخل الاجراء.

قيمة المعامل في هذا النوع تكون محمية، أي أنها تحتفظ بقيمتها خارج الاجراء (لا تتغير)، بمعنى الإجراء يأخذ نسخة من هذه القيمة ليستفيد منها داخل الاجراء ولا يقوم بتغييرها.

مثال: إنشاء إجراء مخزن باسم عرض_الزبون يقوم بعرض بيانات الزبون من جدول الزبون باستخدام معامل إدخال IN. أنظر الشكل

```
DELIMITER //
CREATE  PROCEDURE  عرض_الزبون(IN CUST INT)
 Begin
        SELECT * FROM  الزبونWHERE رقم_الزبون = CUST ;
 End //
DELIMITER ;
```

معامل الإدخال IN Parameter

```
DELIMITER //
CREATE  PROCEDURE  عرض_الزبون(IN CUST INT)
  Begin
        SELECT * FROM  الزبونWHERE رقم_الزبون = CUST ;
  End //
DELIMITER ;
```

يتم استدعاء الإجراء (عرض_الزبون) كما في الشكل التالي:

```
CALL عرض_الزبون(300) ;
```

ناتج استدعاء الإجراء المخزن في الشكل السابق يظهر في الجدول

رقم الزبون	اسم الزبون
300	عبد المعز خيري

معامل الإدخال IN Parameter

مثال: إنشاء إجراء مخزن باسم حذف_مالك_عقار يقوم بحذف بيانات مالك عقار من جدول مالك العقار باستخدام معامل إدخال IN. أنظر الشكل

```
DELIMITER //
CREATE  PROCEDURE  حذف_مالك_عقار(IN CUST INT)
 Begin
      DELETE FROM  مالك_العقارWHERE رقم_مالك_العقار = CUST;
 End //
DELIMITER ;
```

يتم استدعاء الإجراء المخزن (حذف_مالك_عقار) كما في الشكل.

```
CALL حذف_مالك_عقار(2) ;
```

معامل الإخراج OUT Parameter

يستخدم معامل الإخراج OUT Parameter مع الاجراء لإرجاع قيمة من داخل الاجراء. بمعنى يتم تنفيذ جملة الاستعلام داخل الاجراء، ونتيجة جملة الاستعلام يتم تخزينها في هذا المعامل OUT، ويتم ترجيع القيمة إلى خارج الاجراء، ويتم الاستفادة من هذه القيمة في أي برنامج خارجي قام باستدعاء الإجراء أو يتم عرض القيمة خارج الاجراء.

معامل الإخراج OUT Parameter

مثال: إنشاء إجراء مخزن باسم عدد_الإيجارات، حيث يقوم بعرض عدد الإيجارات لزبون معين.

```
DELIMITER $$
CREATE  PROCEDURE عدد_الإيجارات(IN C1 INT , OUT O1 INT)
Begin
SELECT COUNT(رقم_الملكية) INTO O1 FROM التأجير WHERE رقم_الزبون = C1;
  End $$
DELIMITER ;
```

لاستدعاء الإجراء بطريقة صحيحة، يتم تعريف متغير Variable خارجي، يتم وضع اسم المتغير داخل أقواس الإجراء ليحل محل المعامل OUT بالإضافة إلى قيمة رقمية لتحل محل معامل IN داخل الإجراء.

OUT Parameter معامل الإخراج

```
DELIMITER $$
CREATE  PROCEDURE عدد_الإيجارات(IN C1 INT , OUT O1 INT)
Begin
SELECT COUNT(رقم_الملكية) INTO O1 FROM التأجير WHERE رقم_الزبون = C1;
  End $$
DELIMITER ;
```

لاستدعاء الإجراء بطريقة صحيحة، يتم تعريف متغير Variable خارجي، يتم وضع اسم المتغير داخل أقواس الإجراء ليحل محل المعامل OUT بالإضافة إلى قيمة رقمية لتحل محل معامل IN داخل الإجراء.

جدول التأجير

رقم_الزبون	رقم_الملكية	تاريخ_التأجير	تاريخ_النهاية
100	10	2018-01-01	2018-06-30
100	20	2018-07-01	2018-12-31
200	10	2018-07-01	2018-12-31
200	30	2019-07-01	2019-12-31
200	20	2020-01-01	

```
DECLARE  X INT ;
CALL عدد_الإيجارات(200 , @X) ;
SELECT @X;
```

معامل الإدخال والإخراج INOUT Parameter

يستخدم معامل الإدخال والإخراج INOUT Parameter مع الاجراء ليتم إدخال قيمة للإجراء وترجيع قيمة في نفس المعامل Parameter خارج الإجراء، هذا يعني أنه مدمج بين النوعين IN وOUT. بمعنى يتم تمرير قيمة لداخل الاجراء، قيمة المعامل تتغير من جديد داخل الاجراء، ويتم ترجيع القيمة الجديدة إلى خارج الإجراء ليتم الاستفادة منها في أي برنامج استدعاء الإجراء أو يتم عرضها خارج الإجراء.

INOUT Parameter معامل الإدخال والإخراج

مثال: إنشاء إجراء مخزن باسم مجموع_الإيجارات يقوم بعرض مجموع الإيجارات لمالك العقار.

```
DELIMITER &&
CREATE PROCEDURE مجموع_الإيجارات (INOUT SUM1 INT(6))
 Begin
      DECLARE S1 INT;
      Select SUM(الإيجار_الشهري) INTO S1 from الملكية WHERE رقم_مالك_العقار =
SUM1 GROUP BY رقم_مالك_العقار;
      SET SUM1 = S1;
 End &&
DELIMITER ;
```

جدول الملكية

رقم_الملكية	عنوان_الملكية	الإيجار_الشهري	رقم_مالك_العقار
10	السراج	1500	1
20	بن عاشور	1000	2
30	قرقارش	2000	2
40	زناته	1500	3

INOUT Parameter معامل الإدخال والإخراج

```
DELIMITER &&
CREATE  PROCEDURE مجموع_الإيجارات (INOUT SUM1 INT(6))
 Begin
      DECLARE S1 INT;
      Select SUM(الإيجار_الشهري) INTO S1 from الملكية WHERE رقم_مالك_العقار =  SUM1
GROUP BY رقم_مالك_العقار;
      SET SUM1 = S1;
 End &&
DELIMITER ;
```

جدول الملكية

رقم_الملكية	عنوان_الملكية	الإيجار_الشهري	رقم_مالك_العقار
10	السراج	1500	1
20	بن عاشور	1000	2
30	قرقارش	2000	2
40	زناته	1500	3

يتم استدعاء الإجراء مجموع_الإيجارات كما في الشكل

```
DECLARE  X INT(6) ;
SET @X = 2 ;
CALL مجموع_الإيجارات(@X) ;
SELECT @X AS "مجموع الايجارات لمالك العقار" ;
```

مجموع الايجارات لمالك العقار
3000

الجملة الشرطية IF Statement

أحيانا نحتاج إلى إجراء مقارنات داخل الإجراء قبل ترجيع أي قيمة إلى خارج الإجراء، يمكن ذلك باستخدام جملة IF الشرطية.

تُستخدم الجملة الشرطية IF لمقارنة القيم داخل الإجراء المخزن التي تم الحصول عليها من جملة الاستفسار (الاسترجاع) SELECT.

الشكل العام لتركيب جملة IF كما في الشكل:

```
IF (condition1) THEN  <SQL statement1>
[ELSIF (condition2) THEN <SQL statement2>]
[ELSE <SQL statement3>]
END IF;
```

الجملة الشرطية IF Statement

مثال: إنشاء إجراء مخزن باسم سعر_الإيجار يقوم بمقارنة سعر الإيجار الشهري في جدول الملكية، حيث يتم تمرير رقم الملكية للإجراء، تقوم جملة الاستفسار بالبحث على رقم الملكية ومقارنة سعر الإيجار الشهري الخاص بالملكية حسب الشروط ثم ترجيع نص معين، إذا كان الإيجار الشهري أقل من 1000 يرجع الاستفسار نص "الإيجار رخيص"، إذا كان الإيجار الشهري ما بين 1000 و1500 يرجع الاستفسار نص "الإيجار مرتفع"، إذا كان الإيجار الشهري أكبر من 1500 يرجع الاستفسار نص "الإيجار مرتفع جدا". ويتم عرض النص خارج الإجراء فيما بعد.

IF Statement الجملة الشرطية

```
DELIMITER $$
CREATE PROCEDURE سعر_الإيجار ( IN SS1 INT(2), OUT S2 varchar(20) )
BEGIN
   DECLARE X1 double;
   SELECT الإيجار_الشهري INTO X1 FROM الملكية WHERE رقم_الملكية = SS1;
   IF X1 < 1000 THEN  SET S2 =  "الإيجار رخيص";
   ELSEIF (X1 >=1000 AND X1 <= 1500) THEN  SET S2 =  "الإيجار مرتفع";
   ELSEIF X1 > 1500 THEN SET S2 = "الإيجار مرتفع جدا";
   END IF;
END$$
DELIMITER ;
```

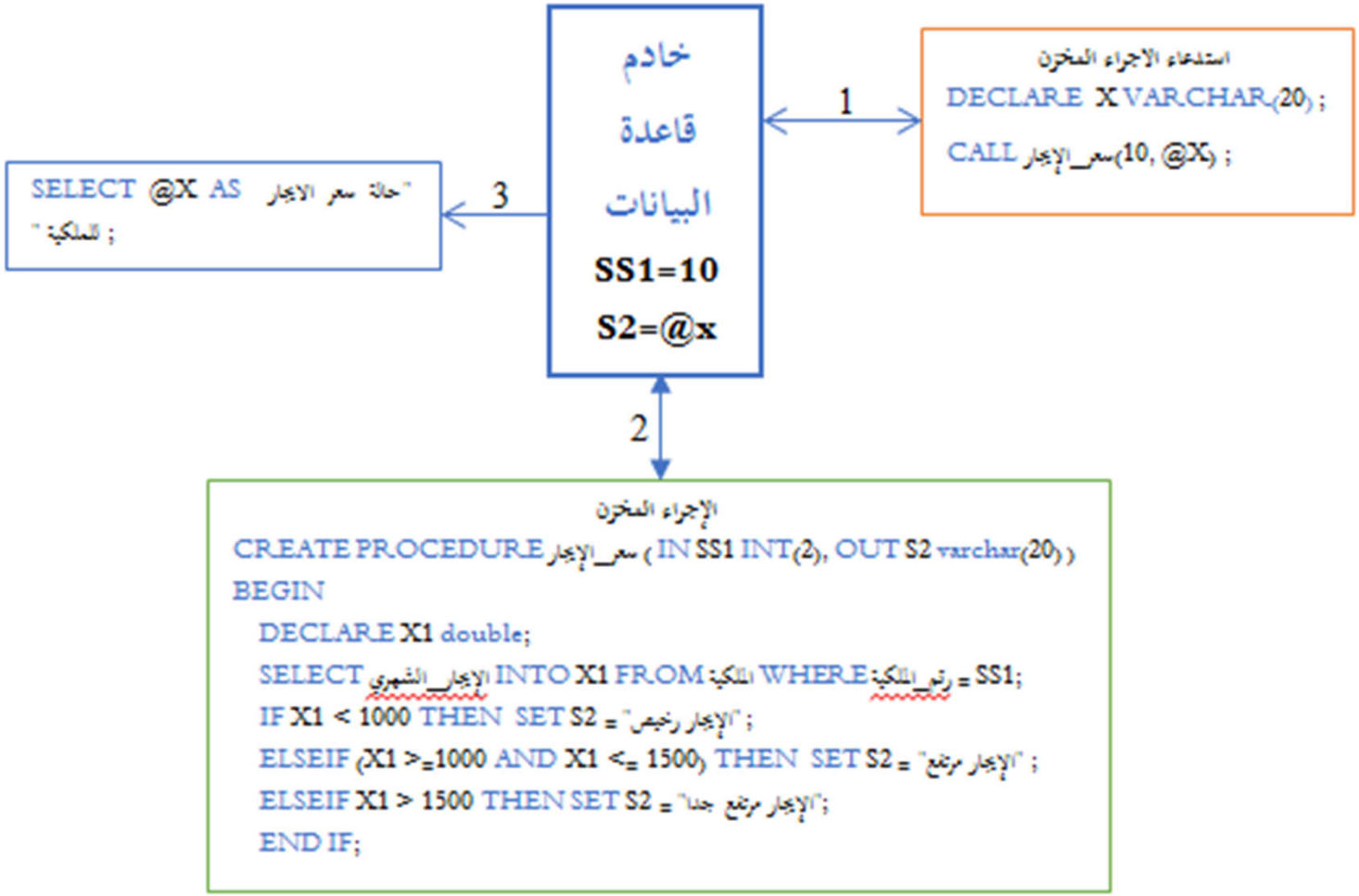
يتم استدعاء الإجراء سعر_الإيجار كما في الشكل

```
DECLARE  X VARCHAR(20) ;
CALL سعر_الإيجار(10, @X) ;
SELECT @X AS " حالة سعر الايجار للملكية" ;
```

حالة سعر الايجار للملكية
الإيجار مرتفع

الجملة الشرطية IF Statement

الشكل يبين كيف يتم استدعاء الإجراء المخزن سعر_الإيجار.



الجملة الشرطية IF Statement

يمكن معرفة الإجراءات المخزنة داخل قاعدة البيانات، لعرض الاجراءات المخزنة نستخدم الأمر في الشكل.

```
SHOW  PROCEDURE  STATUS ;
```

لحذف الإجراء المخزن مجموع_الإيجارات من قاعدة البيانات نستخدم الأمر في الشكل التالي:

```
DROP PROCEDURE مجموع_الإيجارات;
```

Summary ملخص

- في البداية تم توضيح كيفية معالجة المعاملات Transaction Processing التي تدير وتراقب اتمام جميع العمليات على الجداول العلائقية بطريقة سليمة دون حدوث أخطاء لضمان سلامة البيانات. تقوم عملية المعالجة على عمليات الادخال INSERT والحذف DELETE والتعديل UPDATE فقط، يتم ذلك باستخدام جملتي الاعتماد COMMIT والتراجع ROLLBACK.
- تقوم جملة ROLLBACK باسترجاع ما كانت عليه الجداول قبل تنفيذ عمليات الادخال، والحذف والتعديل، أما جملة COMMIT تقوم باعتماد تخزين نتائج جميع العمليات السابقة التي تم تنفيذها. كل نظام إدارة قواعد البيانات DBMS لديه طريقة في التعامل مع Transaction.
- بعد ذلك انتقلنا إلى الإجراء المخزن STORED PROCEDURES الذي يُخزن داخل قاعدة البيانات مع الجداول. عند استدعاء الإجراء باستخدام أمر الاستدعاء CALL يقوم بعرض بيانات الجداول المرتبط بها، كما يمكن أن يستخدم في عمليات التحديث على الجداول مثل إدخال بيانات أو حذف بيانات أو تعديل في البيانات. نستطيع داخل الاجراء التصريح على متغيرات Variables وتعريف معاملات Parameters لإدخال أو إخراج قيم للإجراء. بالإضافة إلى امكانية استخدام جملة IF الشرطية لمقارن القيم داخل الاجراء.

نهاية المحاضرة