



جامعة المنارة

كلية: الهندسة

قسم: المعلوماتية

اسم المقرر: قواعد بيانات ٢

رقم الجلسة (١٠)

عنوان الجلسة

تمارين عن المناظير و المؤشرات

د. ايهاب ديباجة

م. جهاد عيسى

م. جمال محمود

العام الدراسي: ٢٠٢٥-٢٠٢٦

الفصل الدراسي : الأول

جدول المحتويات

Contents

رقم الصفحة	العنوان
٣	المناظير Views
٤	المؤشرات Cursors
٤	مثال عملي

الغاية من الجلسة:

- ✓ بناء المناظير VIEWS
- ✓ بناء المؤشرات Cursors
- ✓ أمثلة عملية

المناظير VIEWS

هي تعابير Select محفوظة تسمح للمستخدم بالعمل مع النتائج التي تعيدها و هي غالباً ما تستخدم كطريقة لتغليف الاستعلامات أو عمليات الدمج المعقدة بين الجداول و حفظ التعليمات ضمن قاعدة البيانات بحيث يمكن استدعاء التعليمات من خلال استدعاء المنظار. كما أنها تعد من أهم أدوات الأمان لأنها تساعد على منع المستخدمين ذوي الصلاحيات المحدودة من الوصول إلى الجداول الأصلية في قواعد البيانات و منحهم صلاحيات قراءة فقط أو منحهم صلاحيات لرؤية لبعض الحقول من الجدول فقط كما تساهم المناظير في اخفاء اسماء الحقول الحقيقية في الجداول الاساسية.

صيغة إنشاء منظار

```
CREATE VIEW view_name AS select_statement
```

نلاحظ أننا يجب أن نعطي المنظار اسماً محدداً بالوصفة view_name ومن ثم نكتب الاستعلام المغلف بهذا المنظار بعد الكلمة المفتاحية as.

مثلاً، لتغليف الاستعلام الذي يعيد اسم وبلد كل زبون في الجدول Customers من القاعدة Northwind فإننا نكتب الأمر التالي:

```
USE northwind
```

```
Go
```

```
CREATE VIEW v_c
```

```
AS SELECT CompanyName, Country FROM Customers
```

نستطيع الآن الاستعلام عن السجلات الناتجة عن المنظار كما في حالة الجداول باستخدام select. مثلاً لتحديد قائمة الزبائن الذين يحوي اسم بلدهم على الحرف u نستخدم الاستعلام التالي:

```
select * from v_c where country like '%u%'
```

صيغة تعديل منظار
نستخدم التعليمة التالية

```
Alter VIEW view_name AS NEW_select_statement
```

مثلاً لتعديل المنظار السابق بحيث نضيف اسم مدينة الزبون نكتب الأمر التالي:

```
USE northwind
```

```
Go
```

```
Alter VIEW vwCustomerCountry
```

```
AS SELECT CompanyName, Country, City FROM Customers
```

صيغة حذف منظار
لحذف منظار نستخدم التعليمة التالية

```
DROP VIEW view_name
```

المؤشرات CURSORS

تسمح المؤشرات بالمرور على نتائج تعليمة **Select** سطرًا سطرًا، ومن أجل كل سطر من الأسطر يمكن إجراء معالجة محددة. الصيغة النظامية لتعريف مؤشر هي على الشكل التالي:

```
DECLARE cursor_tic
FOR select_statement
```

ملاحظات: بعد تعريف أي مؤشر :

١. يجب فتح المؤشر قبل استخدامه:

Open cursor_name

٢. يتم إحضار السجل التالي باستخدام التعليمة :

Fetch next from cursor_name

٣. يتم إحضار السجل السابق باستخدام التعليمة:

Fetch prior from cursor_name

٤. يتم إحضار السجل الأول باستخدام التعليمة:

Fetch first from cursor_name

٥. يتم إحضار السجل الأخير باستخدام التعليمة:

Fetch last from cursor_name

٦. عند إحضار سجل باستخدام إحدى التعليمات السابقة يجب دوماً فحص قيمة المتحول **@@FETCH_STATUS**، فإذا كانت -٢ هذا يعني أن السجل لا يمكن إحضاره إما لأنه قد عدل أو لأنه قد حذف.

٧. تنتهي عملية إحضار السجلات عندما تصبح قيمة المتحول **@@FETCH_STATUS < > 0**.

(0 يعني نجاح عملية الإحضار، ١- فشل أو أننا خرجنا خارج المؤشر، -٢ يعني السجل محذوف)

٨. عند الانتهاء من استخدام المؤشر يجب تحرير المؤشر باستخدام التعليمات التالية:

Close cursor_name

Deallocate cursor_name

أمثلة على المناظير و المؤشرات

سنقوم ببناء قاعدة بيانات باسم **V_cursors** ثم بناء جدول خاص بالطلاب و جدول خاص بالمواد و جدول علامات الطلاب و أخيراً جدول نتائج الطلاب (ناجح - راسب) و سنقوم بانشاء منظار نظهر فيه اسم الطالب و اسم المادة و علامة الطالب بالمادة المحددة ثم مؤشر **cursor** يقوم بتخزين رقم الطالب و وضعه بعد الانتهاء من تقديم المواد بحيث يتم تحديد وضع الطالب ناجح **success** إذا كان عدد المواد التي حقق فيها حد النجاح و هو ٦٠ درجة في ثلاث مواد على الأقل و إلا يعتبر راسب **Fail**

Create database v_cursors

Use v_cursors

```
create table stud_name (stud_id int identity(1,1) primary key,f_name varchar(50),l_name varchar(50))
```

```
create table subject(sub_id int identity(1,1) primary key,sub_name varchar(40))
```

```
create table stud_mark (sub_id int foreign key references subject (sub_id),stud_id int foreign key references stud_name (stud_id),exam_mark int)
```

```
Create table final_res(stud_id int,resault varchar(40))
```

```
insert into stud_name values('samer','kasem'),('lana','safi'),('nada','yousef'),('ramiz','sofi')
```

```
insert into subject values('programming 1'),(' networking 1'),('database1'),('operating system 1')
```

```
insert into stud_mark values
```

```
(1,1,66),(1,2,45),(1,3,73),(1,4,55),
(2,1,72),(2,2,62),(2,3,37),(2,4,41),
(3,1,88),(3,2,63),(3,3,79),(3,4,36),
(4,1,66),(4,2,77),(4,3,52),(4,4,78)
```

```
create table stud_name (stud_id int identity(1,1) primary key,f_name varchar(50),l_name varchar(50))
```

```
create table subject(sub_id int identity(1,1) primary key,sub_name varchar(40))
```

```
create table stud_mark (sub_id int foreign key references subject (sub_id),stud_id int foreign key
references stud_name (stud_id),exam_mark int)
```

```
insert into stud_name values('samer','kasem'),('lana','safi'),('nada','yousef'),('ramiz','sofi')
```

```
insert into subject values('programming 1'),(' networking 1'),('database1'),('operating system 1')
```

```
insert into stud_mark values
```

```
(1,1,66),(1,2,45),(1,3,70),(1,4,55),
(2,1,68),(2,2,66),(2,3,55),(2,4,54),
(3,1,76),(3,2,77),(3,3,59),(3,4,73),
(4,1,86),(4,2,84),(4,3,72),(4,4,66)
```

١- المطلوب بناء منظار view يقوم بعرض اسم المادة و اسم و كنية الطالب كعامود واحد يعبر عنه بالاسم الكامل و علامة الطالب

```
create view exam_marks as
select sub_name,concat(f_name,' ',l_name) as full_name,exam_mark
from stud_mark sm join subject su
on sm.sub_id = su.sub_id
join stud_name sn
on sm.stud_id = sn.stud_id
```

٢- بناء اجرائية تقوم بتسجيل رقم الطالب و حالته (ناجح-راسب) بناءً على عدد المواد التي نجح فيها بحيث يسجل الطالب ناجح في حال نجح بعدد مواد أكثر من اثنين و راسب في حال غير ذلك

```
CREATE PROCEDURE FinalResult
as begin
declare @res1 varchar(40);
declare @sum_suc int;
declare @id1 int;
declare stud_id1 cursor for
select stud_id from stud_name;
open stud_id1 ;
```

```
fetch next from stud_id1 into @id1;
while @@fetch_status = 0
begin
select @sum_suc = count(*) from stud_mark where stud_id = @id1 and exam_mark > 59;
if @sum_suc > 2
set @res1 = 'success' ;
else
set @res1 = 'fail' ;
insert into final_res values(@id1,@res1);
fetch next from stud_id1 into @id1;
end ;
close stud_id1;
deallocate stud_id1;
end ;
```