**قم بربط قاعدة البيانات Teaching لديك على الحاسب وقم بدراسة وتنفيذ الاستعلامات التالية:**

**إدخال البيانات:**

insert into card\_info (card\_id,card\_number,card\_pass,card\_expire,card\_balance) values (1,'1111','1111',null,null)

insert into card\_info (card\_id,card\_number,card\_pass,card\_expire,card\_balance) values (2,'2222','2222',null,null)

insert into student (std\_id,std\_fname,std\_lname,std\_mail,std\_birthdate) values (1,'xxxx,null,null,null)

insert into student (std\_id,std\_fname,std\_lname,std\_mail,std\_birthdate) values (2,'student\_123456',null,null,null)

insert into term (term\_id,term\_name) values (1,'f15')

insert into term (term\_id,term\_name) values (2,'s15')

insert into course (course\_id,course\_name,course\_code,course\_program\_code) values (1,'db',db,1)

insert into course (course\_id,course\_name,course\_code,course\_program\_code) values (2,'db',db,1)

insert into student\_profile (std\_id,course\_id,term\_id) values (1,1,1)

insert into student\_profile (std\_id,course\_id,term\_id) values (1,2,1)

1. عرض الفواتير المدفوعة من قبل اي طالب اسمه الاول سامر في الفصل s23

طريقة اولى

select count(\*) from payment where pay\_term\_id =(select term\_id from term where term\_name='s23')

and pay\_std\_id in (select std\_id from student where std\_fname like '%smaer%')

and pay\_ispaid=1

طريقة ثانية

select count(\*) from payment inner join term

on payment.pay\_term\_id=term.term\_id

inner join student

on student.std\_id=payment.pay\_std\_id

where term\_name='s23' and std\_fname like '%samer%'

1. عدد الفواتير المدفوعة في الفصل s23
	1. طريقة اولى

select count(\*) from payment where pay\_ispaid=1 and pay\_term\_id =(

select term\_id from term where term\_name='s23')

* 1. طريقة ثانية

select count(\*) from payment inner join term

on payment.pay\_term\_id=term.term\_id

 where term\_name='s23' and pay\_ispaid=1

1. اسماء المقررات التي قام الطالب الذي اسمه الاول سامر واسمه الاخير عمار بتسجيلها في الفصل s23 وبالتالي دفع عليها دفعات مؤكدة.

select course\_name from course inner join student\_profile

on course.course\_id=student\_profile.course\_id

inner join student

on student.std\_id=student\_profile.std\_id

inner join payment

on payment.pay\_std\_id=student.std\_id

inner join term

on student\_profile.term\_id=term.term\_id

where std\_fname='saner' and std\_lname='ammar'

and payment.pay\_ispaid=1

and term.term\_name='s23'

**قم بحل التمرين باستخدام الاستعلامات الفرعية**

1. رقم البطاقة التي دفع منها الطالب سامر عمار لتسجيل مقرر c++ في الفصل s23

select card\_number from card\_info where card\_id=(

select pay\_card\_id from payment where payment.pay\_id =

(

select pay\_id from payment\_course\_details where course\_id =

(select course\_id from course where course\_name='c++')

) and payment.pay\_std\_id=(select student.std\_id from student where std\_fname='samer' and std\_lname='ammar')

)

**قم بحل التمرين باستخدام inner join**

1. قم بعرض جميع البطاقات مع رقم الدفعات عليها (يشمل العرض حتى البطاقات التي ليس عليها دفعات)

**يجب حل التمرين باستخدام Outer join**

1. قم بعرض اسماء المقررات التي عليها اكثر من 10 فواتير مدفوعة في الفصل s23

طريقة اولى

select term\_name from term where term\_id in (

select pay\_term\_id from (

(select count(\*) as c,pay\_term\_id from payment where pay\_ispaid=1 group by pay\_term\_id

having count(\*)>10)) as tbl)

and term.term\_name='s23'

1. عرض الفصول مجموع الدفعات التي تم دفعها في كل فصلمرتبة ابتداء من أكبر دفعة.

select sum(pay\_balance),term\_name from payment inner join term

on payment.pay\_term\_id =term.term\_id

group by term\_name

order by sum(pay\_balance) desc

1. عرض معلومات كل طالب (الاسم، الكنية، الايميل)(لجميع الطلاب سواء كان له دفعة أم لاء) مع قيمة مجموع دفعاته إن وجدت من اجل كل نوع.

*الحصول على دفعات الطالب حسب النوع من خلال استعلام فرعي يقوم بتجميع الدفعات وفق رقم الطالب ثم الربط مع جدول الطلاب من اجل الحصول على معلومات الطلاب:*

 *قمنا باستعمال الربط الخارجي من اجل الحصول على معلومات كامل الطلاب مع الربط مع استعلام مجموع الدفعات لكل طالب وحسب نوع الدفعة*

SELECT student.std\_fname, student.std\_lname, std\_mail, tbl.sum\_pay, tbl.pay\_type

FROM student FULL OUTER JOIN

 (SELECT SUM(pay\_balance) AS sum\_pay, pay\_std\_id, pay\_type

 FROM payment

 GROUP BY pay\_std\_id, pay\_type) AS tbl ON student.std\_id = tbl.pay\_std\_id

1. عرض اسماء المواد جميعها مع ذكر أرقام وأسماء الطلاب المسجلين إن وجدوا.

قمنا باستخدام الجداء الخارجي من اجل عرض جميع المواد حتى لو كان الطلاب غير مسجل بالمادة

SELECT course.course\_name, student.std\_fname, student.std\_lname

FROM payment\_course\_details FULL OUTER JOIN

 course ON payment\_course\_details.course\_id = course.course\_id INNER JOIN

 payment ON payment\_course\_details.pay\_id = payment.pay\_id INNER JOIN

 student ON payment.pay\_std\_id = student.std\_id

1. قائمة بجميع المواد مع ذكر مجموع ما تم دفعه من تسجيلات على هذه المادة.

SELECT course.course\_name, SUM(payment.pay\_balance) AS Expr1

FROM course INNER JOIN

 payment\_course\_details ON course.course\_id = payment\_course\_details.course\_id INNER JOIN

 payment ON payment\_course\_details.pay\_id = payment.pay\_id

GROUP BY course.course\_name

1. قائمة بأسماء الطلاب الذين لم يقوموا بأي عملية دفع خلال فصل F15

*الحصول على الطلاب الذين ليس لهم دفعات في عام F15 من خلال استعلام فرعي يقوم بالحصول على الطلاب الذين دفعوا في الفصل المطلوب ثم الحصول على معلومات الطلاب الذين لم ترد ارقامهم في الاستعلام السابق.*

 select \* from student where std\_id not in(

 select pay\_std\_id from payment where Pay\_term\_id =(select term\_id from term where term\_name='F15'))

 **كتابة إجرائية تقوم بدفع فاتورة معينة لطالب بحيث: (متحولات الدخل: هي رقم البطاقة، رقم الطالب، رقم الفاتورة المطلوب دفعها)، متحول خرج يطبع رسالة تدل على نجاح أو فشل العملية.**

يجب أن تقوم الإجرائية بالتحقق من كفاية الرصيد، وأن الدفعة ليست مدفوعة سابقا، ثم تقوم بإدراج المواد في جدول مواد الطالب في حال كانت الفاتورة "تسجيل على مواد"

تقوم بتغيير حالة الفاتورة إلى مدفوعة

قم باستدعاء الاجرائية وتنفيذ الإجرائية.

*الإجراء المخزن المطلوب يقوم بالحصول على المبلغ الموجود في البطاقة و المبلغ المطلوب في الدفعة والمقارنة وبنفس الشرط يقوم بالتحقق من إن الفاتورة لم يتم دفعها من قبل وفي حال كانت مدفوعة يتم طباعة رسالة خطأ وإلا يتم تعديل حالة الفاتورة وإضافة المادة إلى جدول بروفايل الطالب:*

create procedure paycard(@pay\_id int ,@card\_id int ,@std\_id int )

as

declare @bal int

declare @ispaid int

declare @pay\_type varchar(10)

declare @termid int

select @bal =pay\_balance,@ispaid =Pay\_isPaid,@pay\_type =Pay\_type,@termid=pay\_term\_id from payment where pay\_id=@pay\_id

declare @cardbal int

select @cardbal=card\_balance from card\_info where card\_id=@card\_id

if (@bal>@cardbal ) or (@ispaid =1)

 print ('error')

 else

 begin

 update payment set Pay\_isPaid =1 where Pay\_id =@pay\_id

if (@pay\_type='registeration')

begin

insert into student\_profile select @std\_id,course\_id,@termid from payment\_course\_details where Pay\_id =@pay\_id

end

END

*تنفيذ الاجراء المخزن:*

**

**كتابة قادح يقوم بـإدراج سطر جديد في جدول Payment\_log عند تغيير حالة الدفعة ispaidأي في كل مرة نقوم بعملية دفع. قم بعملية تتطلب استدعاء القادح وتنفيذه**

*القادح المطلوب يقوم بالحصول على معلومات الفاتورة التي تم تعديلها ومن ثم يقوم بإضافة القيم المطلوبة إلى سجل الدفعات*

create trigger log\_trig on payment for insert

as

declare @payid int

declare @pay\_type varchar(10)

select @payid =pay\_id,@pay\_type=pay\_type from inserted

insert into payment\_log(pay\_id,date\_time,pay\_type) values(@payid ,GETDATE(),@pay\_type)