الجمهورية العربية السورية
وزارة التعليم العالي
جامعة حمص
الكلية التطبيقية
قسم تقنيات حاسوب

قواعد معطيات 1

القسم العملي

المحاضرة الرابعة

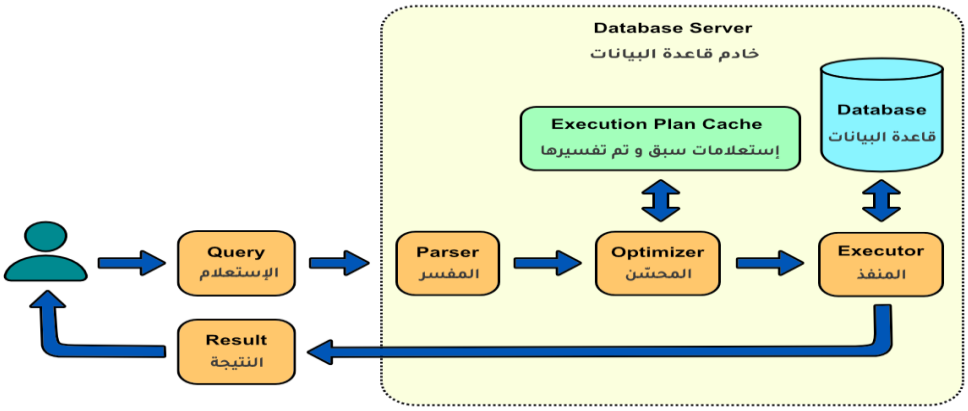
Structure Query Language

(SQL)

اعداد المدرسين:

م . زينب مراد

م . بتول الليوس



SQL Query ?

بعد إنشاء وتنظيم البيانات بقاعدة المعطيات ، سيكون هنالك حاجة لطلب هذه البيانات.

فعندما نريد استرجاع أو استعلام عن بيانات ما سواء كان هذا الاستعلام معقد أو بسيط، فهذه العملية تسمى (**Query**).

لغة استعلام المعطيات

DATA QUERY LANGUAGE (DQL)

تعرف أيضا" باسم لغة استرجاع البيانات Data Retrieve Language (DRL)

الغرض الأساسي منها هو تحديد واسترجاع البيانات من المخطط(الجداول, العروض, ...) بناء على الاستعلامات, وتتيح للمستخدمين فرض الترتيب, التصفية , والشروط على البيانات المسترجعة .

أهم خصائصها : - **المرونة :** يمكن تخصيص الاستعلامات باستخدام الشروط,الربط,والتجميعات.

- **ترتيب البيانات:** النتائج يمكن فرزها وتصفيتها لتكون أكثر تنظيما" وهيكلية.

- **غير إجرائية :** يحدد المستخدم مايريد من البيانات وليس كيف يتم استرجاعها

الأمر الأساسي فيها :

SELECT : تُستخدم لتحديد البيانات التي نريد استخراجها من قاعدة المعطيات ويسترجع البيانات من جدول أو أكثر

وفقا" للاحتياجات وتتيح استخدام عمليات أو شروط مثل:

(WHERE) التصفية.

(ORDER BY) الترتيب.

(GROUP BY)التجميع . وتتيح أيضا الربط بين الجداول باستخدام تعليمات الربط .

الشكل العام لتعليمة SELECT:

Select [**all**|**distinct**] [**top** exp[**percent**]] <column-list>

[from < source table (s) > **]**

[where < restrictive condition > **]**

[group by < column name or expression using a column in the select list > **]**

[having < restrictive condition based on the group by results > **]**

[order by < column list > [**asc**|**desc**] **]**

حيث :

SELECT : كلمة محجوزة وتعني اختيار وهي الامر الأساسي .

ALL : تعني كل وهي الحالة الافتراضية أي إعادة كل السجلات ويمكن استبدالها بالنجمة * .

Distinct : تعني التخلص من السجلات المكررة الموجودة في جدول أو عدة جداول .

Top exp[percent]:

تعيد n سجل من السجلات المعادة في الاستعلام او يمكن ان نحدد نسبة مئوية من السجلات المعادة باستخدام

الكلمة percent مثلا" نريد 50% من السجلات المعادة أو Top 50 percent.

<column-list> : أسماء الأعمدة التي نريد إعادة قيمها في الاستعلام ويفصل بينها فاصلة واذا اردنا اختيار كل الاعمدة نضع *

From : كلمة محجوزة تعني من وهي كلمة أساسية من جملة الاستعلام نحدد بعدها مكان تموضع البيانات

التي اريد الاستعلام عنها او جلب البيانات منها.

<source table (s)>:اسم الجدول او الجداول التي نريد الاستعلام عنها وجلب البيانات منها.

Where <restrictive condition> : كلمة محجوزة تمكننا من اختيار او استعلام عن سجلات محددة وفق شروط محددة

Group by: تستخدم لتجميع المعلومات حسب عمود معين أو عدة أعمدة أو باستخدام توابع التجميع .

Having: كلمة محجوزة تستخدم لوضع شرط على القيم المجمعة بواسطة توابع التجميع بدلا" من **Where**

Order by < column-list >:كلمة محجوزة تستخدم لترتيب النتيجة المعادة وفق عمود معين أو أكثر من عمود حيث:

[Asc|Desc] : Asc : كلمة محجوزة تعني ترتيب تصاعدي وهو الترتيب الافتراضي في حال لم نحدد نوع الترتيب

Desc : كلمة محجوزة تعني ترتيب تنازلي .

- اذا أردنا إعطاء اسم مستعار يدل على اسم عمود جديد أو اسم عمود موجود سابقا باسم آخر نستخدم الكلمة AS وهي كلمة محجوزة

حيث نكتب اسم التابع وبعدها كلمة AS ثم الاسم المستعار الذي نريد اظهاره في نتيجة الاستعلام وهي تستخدم في استعلامات توابع التجميع.

عرض كل الأعمدة والصفوف الموجودة في الجدول :

```
SELECT * FROM table_name;
```

Example: Select * From Employees;

عرض أعمدة محددة من الجدول :

```
SELECT column1, column2
FROM table_name;
```

Example: Select FirstName,LastName From Employees;

التصفية باستخدام عبارة شرط (Where) : الصيغة العامة :

```
SELECT column1
FROM table_name
WHERE condition;
```

Example: Select FirstName From Employees Where Salary > 10000;

Select FirstName From Employees Where Emp_id = 6;

معاملات المقارنة التي تستخدم في عبارات الشرط أو التصفية :

=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal

ملاحظة :

-- تُستخدم لبدء تعليق في سطر واحد في **SQL** ويكون أي شيء بعدها حتى نهاية السطر تعليقًا

/* ***/** تستخدم لتعليق عدة أسطر , والتعليقات لايتم تنفيذها كجزء من الاستعلام .

***** تستخدم لاسترجاع جميع الأعمدة من الجدول المحدد في الاستعلام .

<> أو **!=** كلاهما تعبر عن عملية لايساوي.

أوامر للبحث حسب النمط تستخدم ضمن عبارة Where :

BETWEEN تستخدم للفلترة بين قيمتين ضمن عبارة الشرط ويتبعها احدى المعاملات المنطقية AND,OR,NOT

```
Example: Select FirstName,Salary From Employees Where Salary BETWEEN 10000 AND 80000;
```

IN تستخدم اذا كانت القيمة المطلوبة ضمن القيم الموجودة في قائمة .

```
Example: Select Emp_id,FirstName,Salary From Employees Where Emp_id IN (1,8,12);
```

LIKE تستخدم للفلترة حسب القيم المحددة : مثلا" :

[ ] يدل على أي محرف يقع بين المحرفين الذين تم تمريرهما مثلا [a-d]تعني ارجاع القيم المحصورة بين a و d أي a,b,c,d

% للدلالة عن وجود عدة محارف. _ تعني وجود محرف واحد فقط.

^ تعني النفي Not أي ان المحرف المذكور سوف يتم استثناؤه .

مثال: لاظهار بيانات الموظفين الذين يبدأ اسمهم الأول بالحرفh

```
Select * From Employees where FirstName like 'h%'
```

لاظهار بيانات الموظفين الذين يبدأ اسمهم الأول بأي حرف من a الى h

```
Select * From Employees where FirstName like '[a-h]%'
```

IS NULL تستخدم لتحديد ان عمود ما لا يحوي قيم أو جدول ما لايحوي بيانات في الاعمدة الموجودة ضمنه .

في حال اضفنا عمود لجدول بدون إضافة بيانات ضمنه يضع مكان القيم كلمة NULL

```
Select FirstName from Employees where salary >10000 and address is null;
```

IS NOT NULLتستخدم لتحديد أن البيانات غير فارغة أو قيم العمود أو الأعمدة الموجودة في جدول لاتحوي قيم فارغة .

الترتيب ORDER BY: لترتيب البيانات الموجودة في أعمدة محددة او في الجدول كامل في قاعدة المعطيات بعد الاستعلام عنها .

ويمكن ترتيب النتائج المعادة بعد الاستعلام **تصاعديا" أو تنازليا"** **ASC, DESC**

- **<u>الترتيب الافتراضي</u>** للبيانات المعادة بدون استخدام الكلمة المحجوزة للترتيب هو **بشكل تصاعدي .**

مثال: لاظهار الاسم الأول للموظفين بدءا" بالراتب الأعلى أولا" نستخدم الترتيب التنازلي :

```
Select FirstName From Employees Oreder by Salary DESC;
```

لاظهار بيانات الموظف ذو الراتب الأعلى وتابع للقسم الأول .

```
Select top(1) * From Employees Where dept_id=1 Order by Salary DESC;
```

لاظهار عمود أسماء الموظفين مرتب ترتيبا" تصاعديا" حسب الترتيب الابجدي.

يمكن كتابة الاستعلام بأحد الشكلين التاليين:

```
Select FirstName From Employees ; -- الحالة الافتراضية ASC

Select FirstName From Employees Order by FirstName ASC ;
```

التجميع باستخدام عبارة GROUP BY :

نستخدم العبارة لتجميع المعلومات من قاعدة المعطيات أو من جدول حسب عمود معين أو عدة أعمدة ويتم التجميع في استعلام واحد باستخدام أحد توابع التجميع Aggregate Functions .

- Aggregate Functions -

- MAX (column-name): يستخدم للحصول على أكبر قيمة في العمود

مثال: استعلام عن اعلى راتب من بين جميع رواتب الموظفين

```
Select  Max( salary ) AS max From Employees;
```

- MIN (column-name): يستخدم للحصول على أصغر قيمة في العمود

مثال: استعلام عن ادنى راتب من بين جميع رواتب الموظفين

```
Select  Min( salary ) AS min From Employees;
```

- SUM (column-name): يستخدم للحصول على مجموع القيم في العمود

مثال: استعلام عن مجموع رواتب الموظفين في كل قسم

```
Select  SUM(salary) AS sum From Employees group by Department_ID;
```

- COUNT (column-name): يستخدم للحصول على عدد القيم في العمود

مثال: استعلام عن عدد الموظفين في كل قسم من الأقسام

```
Select COUNT(Employee_ID) AS count From  Employees  group by  Department_ID;
```

ملاحظة : هنا تم ترتيب القيم في عمود العدد حسب الترتيب الافتراضي للقيم الموجودة في عمود رقم القسم

نتيجة الاستعلام هو عمود واحد فقط وهو عمود العدد COUNT

مثال : استعلام عن رقم القسم وعدد الموظفين فيه

```
Select  Department_ID,COUNT(*)  AS count From Employees  group by Department_ID;
```

ملاحظة : الاختلاف عن المثال السابق هو فقط في النتيجة اظهار عمودين رقم القسم بالإضافة لعمود العدد

- AVG (column-name): يستخدم للحصول على معدل القيم في العمود (متوسط الحسابي = مجموع القيم / عددها)

مثال: استعلام عن معدل رواتب الموظفين

```
Select  AVG(Salary) AS AVG From  Employees;
```

عبارة الشرط HAVING : تستخدم لتطبيق شرط على نتائج التجميع مع Group by بدلا" من where .

مثال : استعلام عن عدد الموظفين الذين معدل رواتبهم اكبر من ٥٠٠٠ في كل قسم من الأقسام .

```
Select COUNT(Employee_id) From Employees group by Department_ID having AVG(salary)>5000;
```

الأمر DISTINCT : كلمة محجوزة تستخدم لأزالة القيم المكررة في عمود .أي تظهر لنا بالنتيجة القيم بدون تكرار

Example: `Select DISTINCT Department_ID From Employees;`

الأمر LIMIT : كلمة محجوزة تستخدم لتحديد عدد النتائج التي نريد اظهارها من قيم العمود او الجدول .

مثال : الاستعلام عن أسماء اول خمس موظفين من جدول الموظفين

```
Select FirstName From Employees LIMIT 5;
```

دوال إضافية لاظهار الوقت والتاريخ :

```
Select DAY('20251110')                    -- يعيد رقم اليوم من التاريخ المعطى بين الاقواس
Select MONTH('20251110')                  -- يعيد رقم الشهر من التاريخ المعطى بين الاقواس
Select YEAR('20251110')                   -- يعيد رقم السنة من التاريخ المعطى بين الاقواس
Select GETDATE()                          -- يعيد الوقت والتاريخ الحالي
Select DATEPART(DAY,GETDATE())            -- النتيجة هي رقم اليوم لان الدالة تعيد جزء من التاريخ على شكل رقم حسب اول وسيط هنا حددنا اليوم
Select DATEPART(MONTH,GETDATE())          -- النتيجة هي رقم الشهر لان الدالة تعيد جزء من التاريخ على شكل رقم حسب اول وسيط هنا حددنا الشهر
Select DATEPART(YEAR,GETDATE())           --النتيجة هي رقم السنة الحالي لان الدالة تعيد جزء من التاريخ على شكل رقم حسب اول وسيط هنا حددنا السنة
Select DATENAME(MONTH,GETDATE())          -- النتيجة هي اسم الشهر لان الدالة تعيد جزء من التاريخ على شكل اسم هنا حددنا الوسيط الاول الشهر
Select DATENAME(WEEKDAY,GETDATE()) -- النتيجة هي اسم اليوم الحالي لان الدالة تعيد جزء من التاريخ على شكل اسم هنا حددنا الوسيط الاول يوم من ايام الاسبوع
```

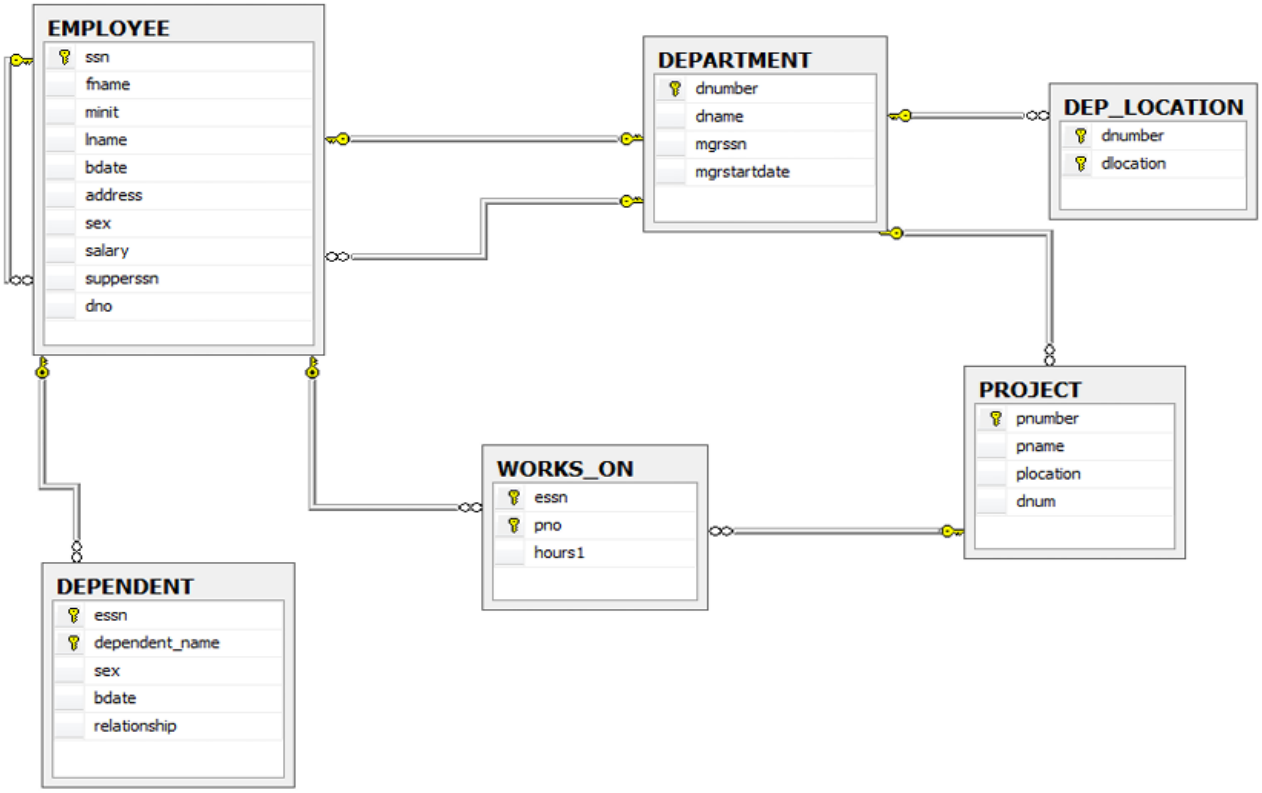
دوال إضافية للتعامل مع السلاسل :

```
Select CHAR(66)                 -- يعيد المحرف المقابل للرقم
Select ASCII('B')               -- يعيد الاسكي المقابل للحرف
Select LOWER('ABCDEFGH')        -- يحول الاحرف الكبيرة الموجودة بين الاقواس الى احرف صغيرة
Select UPPER('abs')             -- يحول الاحرف الصغيرة الموجودة بين الاقواس الى احرف كبيرة
Select LEFT('acvbx',3)          -- يعيد جزء من السلسلة من اليسار حسب الطول المحدد في الوسيط الثاني بين الاقواس
Select RIGHT('ASDFG',4)         -- يعيد جزء من السلسلة من اليمين حسب الطول المحدد في الوسيط الثاني
Select SUBSTRING('abcvfd',3,2)  -- يعيد جزء من السلسلة حسب البداية المحدد في الوسيط الثاني والطول المحدد في الوسيط الثالث
```

ملاحظات :

- عندما نستخدم group by فإن كل عمود من قائمة الأعمدة يجب أن يكون تم تجميعه أو جزء من عبارة group by.
- لايمكن أن نطبق group by على الاسم المستعار للحقل.
- عندما نضع شرط على قيم التجميع يجب استخدام having.
- كل تابع تجميع يتجاهل null، ماعدا التابع count(*).
- للحصول على عدد السجلات غير المكررة count(distinct[col_name])
- يمكن استخدام distinct مع جميع توابع التجميع.
- تطبق دوال التجميع على التعابير ذات القيم أو الاستعلامات الجزئية.
- يمكن استخدام توابع التجميع خارج عملية التجميع.

ليكن لدينا المخطط التالي لقاعدة بيانات شركة ما والمطلوب :



1. إظهار بيانات كل الموظفين .

```
select * from EMPLOYEE
```

2. إظهار اسماء الموظفين وعنوانهم .

```
select fname,lname,address from EMPLOYEE
```

ملاحظة : يمكن اعطاء أي عمود اسم مستعار (alias name)يفيد في اعطاء اسم واضح ومعبر للمستخدم الذي يتلقى نتيجة الاستعلام .

3. إظهار أسماء الموظفين مرتبأ تصاعدياً .

```
select fname as [first name] from EMPLOYEE  order by [first name];

or

select fname  [first name] from EMPLOYEE  order by [first name];
```

ملاحظة: يمكن أن ندمج قيمة عدة أعمدة معا واظهار النتيجة ضمن عمود واحد ومن ثم اعطاءه اسم مستعار.

4. إظهار الاسم الكامل للموظفين.

```
select fname+' '+minit+' '+lname   [full name] from EMPLOYEE
```

ملاحظة: يمكن اجراء أي عملية من العمليات الحسابية (*,/,+,-) على قيم الاعمدة كما يمكن استخدام الأقواس لتحديد الأسبقية.

5. إظهار اسماء الموظفين ورواتبهم قبل الزيادة ورواتبهم بعد زيادة 500

```
select fname+' '+lname   [full name],salary as [befor salary], salary +500 as [new
salary] from EMPLOYEE
```

6. إظهار أرقام الموظفين التابعين لأي مشروع دون تكرار

```
select distinct Essn from WORKS_ON
```

7. إظهار أسماء العشر مشاريع الأولى .

```
select top(10) pname from PROJECT
```

8. إظهار 50% من بيانات الموظفين .

```
select top 50 percent * from EMPLOYEE
```

9. إظهار عناوين الموظفين الذي رواتبهم أكبر أو يساوي 30000

```
select address from EMPLOYEE   where salary >=30000
```

ملاحظة: يمكن وضع اكثر من شرط ضمن عبارة where نربط بينهم باستخدام .not,or,and

10. إظهار بيانات الموظفين الذين رواتبهم بين 10000 و 50000

```
select* from EMPLOYEE where salary between 10000 and 50000
```

11. إظهار أسماء التابعين الذي رقم موظفهم 1 أو 7

```
select dependent_name from DEPENDENT  where essn in(1,7)
or
select dependent_name from DEPENDENT  where essn=1 or essn=7
```

12. إظعار أسماء المشاريع الذين لا يتبعون للقسم 1أو 4

```
select pname from PROJECT where dnum not in (1,4)
```

13. إظهار أسماء الموظفين الذي راتبهم أكبر من 10000 ولا نعرف عنوانهم .

```
select fname from EMPLOYEE where salary >10000 and address is null
```

14. إظهار الاسم الكامل لمدير الشركة.

```
select fname+' '+lname  [manger name] from EMPLOYEE where supperssn is null
```

15. إظهار بيانات الموظف ذو الراتب الأعلى وتابع للقسم الأول

```
select top(1) * from EMPLOYEE where dno=1 order by salary desc
```

16. إظهار بيانات الأقسام الذي يبدأ اسمها بالحرف (h).

```
select * from DEPARTMENT where dname like 'h%'
```

17. إظهار بيانات الأقسام الذي يبدأ اسمها بأي حرف من (a) إلى (h).

```
select * from DEPARTMENT where dname like '[a-h]%'
```

19. إظهار رقم الموظف و معدل الساعات التي يعملها.

```
select essn ,AVG(hours1) as avg from WORKS_ON group by essn
```

20. إظهار رقم القسم وعدد الموظفين فيه.

```
select dno,COUNT(*) as count from EMPLOYEE group by dno
```

21. إظهار رقم كل قسم مع راتب الموظف الاعلى في القسم ومرتبة تصاعدياً حسب الراتب الأعلى.

```
select dno,max(salary ) as max from EMPLOYEE group by dno order by max
```

22. إظهارالراتب الأعلى من بين جميع رواتب الموظفين.

```
select max(salary ) as max from EMPLOYEE
```

23. إظهار مجموع الرواتب في كل قسم.

```
select SUM(salary) from EMPLOYEE group by dno
```

24. إظهار رقم الموظف و معدل الساعات التي يعملها بشرط ان يكون معدل ساعاته أكبر من 5 ساعات.

```
select essn ,AVG(hours1) as avg from WORKS_ON group by essn having AVG(hours1)>5
```

25. إظهار عدد الموظفين الذي معدل رواتبهم أكبر من 20000 في كل قسم من الأقسام.

```
select COUNT(ssn) from EMPLOYEE group by dno having AVG(salary)>20000
```

Any
questions