ما هو SQL (Structured Query Language) ؟

SQL هي لغة كمبيوتر قياسية للوصول إلى قواعد البيانات و معالجتها , لوصف مجموعات من البيانات.

ملاحظة : يعتبر SQL من اللغات الحساسة (sensitive).

ماذا يمكن أن تفعل SQL؟

- تنفيذ استعلامات على قاعدة بيانات
- استرداد البيانات من قاعدة البيانات
- إدراج السجلات في قاعدة البيانات
- إنشاء قواعد بيانات جديدة , جداول جديدة في قاعدة بيانات
- وغيرها ...

ماهو نظام إدارة قواعد البيانات RDBMS (Relational Database Management System) ؟

RDBMS هو أساس SQL , يتم تخزين البيانات في RDBMS على هيئة جداول .

SQL Syntax:

جداول قواعد البيانات :
غالباً تحتوي قاعدة البيانات على واحد أو اكثر من الجداول , يتم تعيين لكل جدول اسم فريد , الجدول يتكون من سجلات (records) وهي الصفوف و المجلات (fields) وهي الأعمدة .

column

Std_ID	Std_Name	Std_Phone	Std_City
1	Noura	0559876525	Riyadh
2	Ammar	0532698714	Riyadh
3	Abdullah	0563987784	Dammam
4	Sara	0550553564	Dammam
5	Reham	0522142182	Riyadh
6	Lama	0536262844	Riyadh

row

الجدول اعلاه يحتوي على ستة سجلات (records) لكل طالب و اربعة اعمدة (Std_Id , Std_Name , Std_phone, Std_City) .

التعامل مع قواعد البيانات (Database):

- إنشاء قاعدة بيانات

يتم إنشاء قاعدة بإستخدام أمر " CREATE DATABASE "

```
CREATE DATABASE Students;
```

- عرض قاعدة البيانات

يتم عرض جميع قواعد البيانات بإستخدام أمر "SHOW DATABASES"

```
SHOW DATABASES; #OR SHOW SCHEMAS;
```

- حذف قاعدة البيانات

يتم حذف قاعدة البيانات عن طريق أمر " DROP DATABASE"

```
DROP DATABASE Students;
```

التعامل مع الجداول (Tables):

(Data Definition Language (DDL : يسمح لك بإجراء عمليات مختلفة على قاعدة البيانات مثل إنشاء ، وتعديل ، وحذف الكائنات.

حين نقوم بتعامل مع الجداول , وإنشائها في قاعدة البيانات لابد ان نتذكر الفرق بين " PRIMARY KEY , FOREIGN KEY " :

- PRIMARY KEY : يحمل قيم فريدة (uniquely) لكل سجل في الجدول , ولايمكن أن يحتوي على قيم خالية (Null values) , يحتوي الجدول الواحد على PRIMARY KEY واحد فقط .
- FOREIGN KEY : هو يشير إلى المفتاح الأساسي (PRIMARY KEY) في جدول آخر.

- إنشاء جدول

يتم إنشاء جدول في قاعدة البيانات عن طريق أمر " CREATE TABLE"

```
CREATE TABLE Stds_info(
     Std_ID int NOT NULL,
     Std_Name varchar(255),
     Std_Phone int,
     Std_City varchar(255),
     PRIMARY KEY (Std_ID)
);
```

يتم كتابة الأمر ثم اسم الجدول المراد إنشائه , بداخلة يتم إدراج و كتابة الأعمدة لهذا الجدول و تحديد نوع البيانات لكل عمود .

حين نريد إضافة عمود جديد للجدول بعد إنشائه نستخدم أمر "ALTER TABLE , ADD"

```
ALTER TABLE Students
ADD Std_Email varchar(255);
```

و أيضاً حين نريد تعيين المفتاح الرئيسي بعد إنشاء الجدول

```
ALTER TABLE Stds_info
ADD PRIMARY KEY (Std_ID);
```

- عرض محتويات الجدول :

يتم عرض محتويات الجدول و إستعراض الأعمدة جميعها بإستخدام أمر "SELECT"

```
SELECT * FROM Stds_info ;
```

" * " هنا تعني الكل (All) إي يتم إرجاع جميع الاعمدة.

أيضاً حين نريد إستعراض بعض الأعمدة من جدول معين , نقوم بتحديد أسماء fields , فمن الجدول أعلاه نريد إستعراض فقط Std_ID , Std_Name

```
SELECT Std_ID , Std_Name FROM stds_info ;
```

- عرض جميع الجداول:

يتم عرض جميع الجداول التي توجد في قاعدة البيانات عن طريق أمر"SHOW TABLES"

```
SHOW TABLES ;
```

- تعديل نوع بيانات العمود في الجدول:

في بعض الأحيان نريد تغيير نوع البيانات لأحد الأعمدة في الجدول , لكن لابد أن نحرص أن هذا التغيير لا يضر البيانات السابقة .

```
ALTER TABLE Stds_info
ALTER COLUMN std_id TYPE CHAR(255)

or

ALTER TABLE Stds_into
MODIFY COLUMN Std_ID CHAR(255);
```

- حذف جدول

قبل أن نتعرف على كيفية حذف جدول بالكامل , نريد حذف عمود معين فقط وذلك عن طريق أمر " DROP COLUMN"

```
ALTER TABLE Stds_info
DROP COLUMN Std_Email ;
```

أما حذف الجدول بالكامل فيتم بإستخدام أمر "DROP TABLE"

```
DROP TABLE Stds_info ;
```

التعامل مع البيانات (Data):

يتم التعامل مع البيانات (Data Manipulation Language) عن طريق DATA CRUD:

- إضافة البيانات :

يتم إدراج سطر أو أسطر في الجدول عن طريق أمر "INSERT INTO , VALUES"

```
INSERT INTO Stds_info (Std_ID , Std_Name , Std_phone, std_city)
VALUES (1,'MAHA',050550,'Riyadh')

     OR

INSERT INTO Stds_info
VALUES (2,'AHMED',050000,'Dhahran')
```

- عرض / قراءة البيانات :

نستطيع عرض جميع البيانات في الجدول عن طريق ما تطرقنا له سابقاً

```
SELECT * FROM stds_info;
```

لكن حين نريد عرض بعض البيانات تحت شرط معين نستخدم أمر "WHERE"

```
SELECT * FROM stds_info WHERE Std_Name = 'MAHA';
```

و حين نريد الاستعلام عن البيانات ,لإرجاع قيم مميزة (مختلفة) فقط أي بدون تكرار , نستخدم أمر "DISTINCT"

```
SELECT DISTINCT Std_City
FROM Stds_info ;
```

- تعديل البيانات :

تم تعديل البيانات التي تم تخزينها في الجدول عن طريق أمر "UPDATE , SET"

```
UPDATE Stds_info
SET Std_Name = 'SARA'
WHERE Std_Name = 'MAHA';
```

- حذف البيانات:

حين نريد حذف جميع البيانات في الجدول نستخدم امر "DELETE"

```
DELETE FROM Stds_info;
```

لكن حين نريد سطر أو أسطر معينة تحت شرط نستخدم "WHERE"

```
DELETE FROM Stds_info
WHERE Std_ID ='1';
```

- البحث في البيانات:

يتم البحث عن نمط معين في عمود ما عن طريق أمر "LIKE"

```
SELECT * FROM Stds_info
WHERE Std_Name LIKE 'N%' ;

     or

SELECT * FROM Stds_info
WHERE Std_Name LIKE '%A' ;
```

"%" هي تمثل جزء أو مقطع من الكلمة .

```
SELECT * FROM Stds_info
WHERE Std_Name LIKE '___A' ;
```

or

```
SELECT * FROM Stds_info
WHERE Std_Name LIKE 'S___' ;
```

"_" هي تمثل حرف فقط من الكلمة .

- معاملات البيانات (Operators):

لدينا ثلاث أنواع للمعاملات وهي :

- Arithmetic Operators
- Comparison Operators
- Logical Operators

- معاملات رياضية (Arithmetic Operators):

هي العمليات الحسابية مثل الجمع "+" , والطرح "-" , والضرب "*" , والقسمة "/" , باقي القسمة "%".

```
SELECT 10 + 5 , 45 - 5 , 5 * 5 , 30 / 5 ;
```

- معاملات المقارنة (Comparison Operators):

تساوي	=
أكبر من	>
أقل من	<
أكبر من أو تساوي	>=
أقل من أو تساوي	<=
لا يساوي	<>

- معاملات منطقية (Logical Operators):

ALL	تكون القيمة TRUE إذا تحققت جميع الشروط
AND	تكون TRUE إذا كانت جميع الشروط المفصولة بـ AND صحيحة
ANY	تكون TRUE إذا تحققت أي قيمة من الشرط
BETWEEN	تكون TRUE إذا كانت القيمة ضمن نطاق المقارنات
EXISTS	تكون TRUE في حالة إرجاع سجل أو أكثر من قيمة الاستعلام
IN	تكون TRUE إذا كانت قيمة المعامل تساوي إحدى القيم
LIKE	يكون TRUE إذا كانت القيمة تطابق نمط
NOT	حين لاتكون قيمة الشرط صحيح يعرض القيم , بمعنى آخر نفي الشرط
OR	TRUE إذا كانت أي من الشروط المفصولة بـ OR صحيحة
SOME	تكون القيمة TRUE في حالة استيفاء أي من قيم الاستعلام الفرعي للشرط

- **الدوال التجميعة (Aggregate functions):**

تقوم الدالة التجميعية بإجراء عملية حسابية على مجموعة من القيم ، وإرجاع قيمة واحدة.

- COUNT()
- AVG()
- SUM()
- MIN()
- MAX()
- GROUP BY
- ORDER BY

تقوم الدالة COUNT () بإرجاع عدد الصفوف التي تطابق معيارًا محددًا.

```
SELECT COUNT(country)
FROM Customers
WHERE country = 'USA';
```

ترجع الدالة AVG () القيمة المتوسطة لعمود رقمي.

```
SELECT AVG(amount)
FROM Orders
WHERE amount < 500;
```

ترجع الدالة SUM () المجموع الإجمالي لعمود رقمي.

```
SELECT SUM(amount)
FROM Orders
WHERE amount < 500;
```

تُرجع الدالة MIN () أصغر قيمة للعمود المحدد.

```
SELECT MIN(amount)
FROM Orders ;
```

ترجع الدالة MAX () أكبر قيمة للعمود المحدد.

```
SELECT MAX(amount)
FROM Orders ;
```

تجمع عبارة GROUP BY الصفوف التي لها نفس القيم من عمود و تُلخصها في صفوف.

```
SELECT COUNT(Customer_ID), Country
FROM Customers
GROUP BY Country;
```

غالبًا ما تُستخدم جملة GROUP BY مع الدالات التجميعية (COUNT () و MAX () و MIN () و SUM () و AVG ()) لتجميع مجموعة نتائج.

تُستخدم الكلمة الأساسية ORDER BY لفرز مجموعة النتائج بترتيب تصاعدي أو تنازلي.

```
SELECT customer_id, first_name ,age
FROM Customers
ORDER BY age  ASC;
```

يتم استخدام ASC للترتيب من الاصغر الى الاكبر

```
or

SELECT customer_id, first_name ,age
FROM Customers
ORDER BY age  DESC;
```

يتم استخدام DESC للترتيب من الاكبر الى الاصغر

تقوم الكلمة الأساسية ORDER BY بفرز السجلات بترتيب تصاعدي افتراضيًا.

- JOIN :

في علم البيانات تعتبر "JOIN" مهمة جداً , وهي تستخدم لدمج صفوف من جدولين أو أكثر ، بناءً على عمود مرتبط بينهما.

- Types of JOINs :

- INNER

إرجاع السجلات التي لها قيم متطابقة في كلا الجدولين.

```
SELECT Orders.Order_ID, Customers.first_name
FROM Orders
INNER JOIN Customers ON Orders.Customer_ID = Customers.Customer_ID;
```

- LEFT

إرجاع كافة السجلات من الجدول الأيسر ، والسجلات المتطابقة من الجدول الأيمن.

```
SELECT Customers.first_name, Orders.Order_ID
FROM Customers
LEFT JOIN Orders ON Customers.Customer_ID = Orders.Customer_ID
ORDER BY Customers.first_name;
```

- RIGHT

إرجاع كافة السجلات من الجدول الأيمن والسجلات المتطابقة من الجدول الأيسر

```
SELECT Customers.first_name,Customers.Customer_ID, Orders.item
FROM Customers
RIGHT JOIN Orders ON Customers.Customer_ID = Orders.Customer_ID
ORDER BY Customers.first_name;
```

- FULL

إرجاع كافة السجلات عند وجود تطابق في الجدول الأيمن أو الأيسر

```
SELECT Customers.frist_Name, Orders.Order_ID
FROM Customers
FULL OUTER JOIN Orders ON Customers.Customer_ID=Orders.Customer_ID
ORDER BY Customers.frist_Name;
```

- UNION

يتم استخدام عامل التشغيل UNION لدمج مجموعة النتائج المكونة من عبارتين SELECT أو أكثر.

- يجب أن تحتوي كل عبارة SELECT داخل UNION على نفس عدد الأعمدة
- يجب أن تحتوي الأعمدة أيضًا على أنواع بيانات مماثلة
- يجب أيضًا أن تكون الأعمدة في كل عبارة SELECT بنفس الترتيب

- UNION ALL:

يحدد عامل التشغيل UNION القيم المميزة فقط افتراضيًا. للسماح بالقيم المكررة ، استخدم UNION ALL