

# SQL\_Review

By: Nourah Almutlag

Ruba

## نظام إدارة قواعد البيانات (DBMS):

نظام إدارة قواعد البيانات (DBMS) هو تطبيق برمجي يتفاعل مع المستخدم والتطبيقات وقاعدة البيانات نفسها لإنشاء البيانات وتحليلها. قاعدة البيانات هي مجموعة منظمة من البيانات.

## يوجد نوعين من نظام إدارة قواعد البيانات:

### • Relational Database Management الـ

تستخدم قواعد البيانات الجداول في تخزين البيانات و علاقاتها. و ايضاً تعتبر static, or fixed schema

### • Non-Relational Database Management System الـ

هنا هي لا تلتزم بالجداول في بياناتها فهي تلجئ الى Document stores و Graph databases و Key-value stores. و تعتبر dynamic schemas

## سوف نركز و نتعامل مع Relational Database Management.

### ما هو نظام إدارة قواعد البيانات (RDBMS (Relational Database Management System ؟

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

### ما هو SQL (Structured Query Language) ؟

لغة الاستعلام الهيكلية Structured Query Language أو SQL هي لغة كمبيوتر لتخزين ومعالجة واسترجاع البيانات المخزنة في قاعدة بيانات, 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 ...

# SQL Syntax:

جداول قواعد البيانات :

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

Table				
Field				
id	ISSN-L	ISSNs	PublisherId	Journal_Title
0	2056-9890	2056-9890	1	Acta Crystallographica Section E Crystallographic Communications
1	2077-0472	2077-0472	2	Agriculture
2	2073-4395	2073-4395	2	Agronomy
3	2076-2615	2076-2615	2	Animals
4	2076-3417	2076-3417	2	Applied Sciences
5	2306-5354	2306-5354	2	Bioengineering
6	2079-7737	2079-7737	2	Books
7	2079-6374	2079-6374	2	Books

Record

Value

## أهم أوامر SQL:

1. DDL: لغة تعريف البيانات - Data Definition Language
2. DML: لغة معالجة البيانات - Data Manipulation Language
3. DCL: لغة التحكم في البيانات - Data Control Language
4. DQL: استعلام البيانات - Data Query Language
5. TCL: لغة التحكم في المعاملات - Transaction Control Language



لغة التحكم في - المعاملات TCL	لغة التحكم في - البيانات DCL	لغة معالجة - البيانات DML	لغة تعريف - البيانات DDL
يتم استخدامه: SELECT ما بناء attribute لتحديد على الشرط الموصوف في وجلب WHERE جملة بياناتها	يتم استخدامه - GRANT لمنح المستخدم امتيازات الوصول إلى قاعدة البيانات	إضافة - INSERT بيانات جديدة إلى قاعدة بيانات	إنشاء - CREATE جدول جديد في قاعدة بيانات
	يتم - REVOKE استخدامه لاستعادة الأذونات من المستخدم	تعديل - UPDATE البيانات الموجودة في قاعدة البيانات	يعدل بنية - ALTER جدول موجود
		يزيل - DELETE البيانات من قاعدة البيانات	يحذف جدولاً من - DROP قاعدة بيانات
			يحذف - TRUNCATE جميع الصفوف من الجدول

## التعامل مع قواعد البيانات (Database):

- إنشاء قاعدة بيانات

يتم إنشاء قاعدة باستخدام أمر " CREATE DATABASE "

CREATE DATABASE Students;

- عرض قاعدة البيانات

يتم عرض جميع قواعد البيانات باستخدام 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):

حين نقوم بتعامل مع الجداول , وإنشائها في قاعدة البيانات لابد ان نتذكر الفرق بين

الـ " PRIMARY KEY, FOREIGN KEY "

- الـ PRIMARY KEY: يحمل قيم فريدة (uniquely) لكل سجل في الجدول , ولا يمكن أن يحتوي على قيم خالية (Null values), يحتوي الجدول الواحد على PRIMARY KEY واحد فقط .
- الـ 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 "



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

- إضافة عمود:

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



- تعيين الـ PK:

أيضاً حين نريد تعيين المفتاح الرئيسي بعد إنشاء الجدول



- **تعديل نوع بيانات العمود:**

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



- **حذف عمود:**

لحذف عمود معين فقط وذلك عن طريق أمر " DROP COLUMN ".



- **حذف جدول:**

حذف الجدول بالكامل فيتم باستخدام أمر " DROP TABLE ".



## **التعامل مع البيانات (Data):**

غالباً ما يتم التعامل مع البيانات عن طريق DATA CRUD.

- **إضافة البيانات :**

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



- عرض / قراءة البيانات :

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



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

أيضاً حين نريد إستعراض بعض الأعمدة من جدول معين , نقوم بتحديد أسماء fields.



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



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



- تعديل البيانات:

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



- حذف البيانات:

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



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



- البحث في البيانات:

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



"%" هي تمثل جزء أو مقطع من الكلمة .



"\_ " هي تمثل حرف فقط من الكلمة .

## معاملات البيانات (Operators):

لدينا ثلاث أنواع للمعاملات وهي :

1. Arithmetic Operators
2. Comparison Operators
3. Logical Operators

### ● معاملات رياضية (Arithmetic Operators):

هي العمليات الحسابية مثل:

- الجمع "+"
- الطرح "-"
- الضرب "\*",
- القسمة "/"
- باقي القسمة "%"

### ● معاملات المقارنة (Comparison Operators):

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

### ● معاملات منطقية (Logical Operators):

تكون القيمة TRUE إذا تحققت جميع الشروط	ALL
تكون TRUE إذا كانت جميع الشروط المفصولة بـ AND صحيحة	AND
تكون TRUE إذا تحققت أي قيمة من الشروط	ANY
تكون TRUE إذا كانت القيمة ضمن نطاق المقارنات	BETWEEN
تكون TRUE في حالة إرجاع سجل أو أكثر من قيمة	EXISTS



الاستعلام	
تكون TRUE إذا كانت قيمة المعامل تساوي إحدى القيم	IN
يكون TRUE إذا كانت القيمة تطابق نمط	LIKE
حين لا تكون قيمة الشرط صحيح يعرض القيم , بمعنى آخر نفي الشرط	NOT
صحيحة OR إذا كانت أي من الشروط المفصولة بـ TRUE	OR
تكون القيمة TRUE في حالة استيفاء أي من قيم الاستعلام الفرعي للشرط	SOME

## الدوال التجميعية (Aggregate functions):

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

تقوم الدالة COUNT ( ) بإرجاع عدد الصفوف التي تطابق معيارًا محددًا.	COUNT ( )
ترجع الدالة AVG ( ) القيمة المتوسطة لعمود رقمي.	AVG ( )
ترجع الدالة SUM ( ) المجموع الإجمالي لعمود رقمي.	SUM ( )
ترجع الدالة MIN ( ) أصغر قيمة للعمود المحدد.	MIN ( )
ترجع الدالة MAX ( ) أكبر قيمة للعمود المحدد.	MAX ( )
تجمع عبارة GROUP BY الصفوف التي لها نفس القيم من عمود و تلخصها في صفوف.	GROUP BY
تستخدم الكلمة الأساسية ORDER BY لفرز مجموعة النتائج بترتيب تصاعدي أو تنازلي.	ORDER BY

سوف نتضح لنا هذه المفاهيم مع الأمثلة و التطبيق:

1. سوف نقوم بإرجاع عدد الـ USA من الـ country:



2. نريد إرجاع متوسط الأعمار:



3. نقوم بإرجاع مجموع الـ amount التي أقل من 500:



4. نريد إرجاع أصغر Author عمراً:



5. الآن نقوم بالعكس, نريد إرجاع أكبر Author عمراً:



6. نريد عدد الـ Authors في كل بلد:



الـ output لتوضيح الفكرة:



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

7. نريد عرض بيانات الـ Authors و لكن بترتيب تصاعدي:



يتم استخدام **ASC** للترتيب من الاصغر الى الاكبر

8. نفس المثال السابق و لكن نريد عرضها بترتيب تنازلي:



يتم استخدام **DESC** للترتيب من الاكبر الى الاصغر

**ملاحظة:** تقوم الكلمة الأساسية ORDER BY بفرز السجلات بترتيب تصاعدي افتراضيًا.

## Joins in SQL

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

**أنواع الـ JOINS:**

- Self-join
- Inner join
- Outer join
  - Left join.
  - Right join
  - Full join
- Union

- Union all
- Intersection
- Minus

لنتعرف على كل نوع و تطبيقه:

سنستخدم الجدولين التاليين لتطبيق أنواع مختلفة من استعلامات.

جدول a-basket



جدول b-basket



### • Self join:

هو join من الجدول الى نفسه (يتم ربط الجدول بنفسه), بإختصار يمكننا القول إنه join بين نسختين من نفس الجدول.

◦ لنقوم بعرض fruits بناءً على مقارنة بينهم في الجدول الاول basket\_a؟



**Result:**



### **الـInner join:**

(تسمى أحياناً simple join) هو الـjoin لجولين أو أكثر تُرجع فقط تلك الصفوف التي تفي بشرط الصلة (join).

- كيف نقوم بضم الجدول الأول (basket\_a) بالجدول الثاني (basket\_b) عن طريق مطابقة القيم الموجودة في العمودين Fruit\_a و fruit\_b



**Result:**



- **الـOuter join:**

ترجع الـouter join جميع الصفوف التي تقي بشرط الـjoin وتعيد أيضًا بعض أو كل هذه الصفوف من جدول واحد لا تقي فيه بعض الصفوف الشرط.

- **الـLeft join:**

إرجاع كافة السجلات من الجدول الأيسر ، والسجلات المتطابقة من الجدول الأيمن.  
■ لنقوم بإرجاع جميع الفواكة من الجدول الأيسر و الفواكة المتطابقة من الجدول الأيمن ؟



الجدول الأول[basket\_a]الجدول الأيسر ويسمى الجدول الثاني[basket\_b]الجدول الأيمن.  
في left join, يبدأ في تحديد البيانات من الجدول الأيسر. يقارن القيم الموجودة في العمود Fruit\_a بالقيم الموجودة في العمود Fruit\_b في الجدول basket\_b.



## Result:



لنفترض اننا نريد تحديد الصفوف من الجدول الأيسر التي لا تحتوي على صفوف متطابقة في الجدول الأيمن ، ذلك يمكن باستخدام الصلة اليسرى مع عبارة WHERE:



**Result:**



○

○ **الـRight join:**

الـright join هي نسخة معكوسة من الـleft join.  
إرجاع كافة السجلات من الجدول الأيمن والسجلات المتطابقة من الجدول الأيسر.





**Result:**



وبالمثل ، يمكننا الحصول على صفوف من الجدول الأيمن لا تحتوي على صفوف متطابقة من الجدول الأيسر عن طريق إضافة عبارة WHERE على النحو التالي:



**Result:**



○ **الـFull join:**

إرجاع كافة ال صفوف من الجدولين, مع القيم "null" اذا لم تفي بالشرط.



**Result:**



لإرجاع الصفوف في الجدول التي لا تحتوي على صفوف متطابقة في الجدول الآخر ، يمكنك استخدام full join مع عبارة WHERE مثل هذه:



**Result:**





- **الـUnion:**

يتم استخدام عامل التشغيل UNION لدمج مجموعة النتائج المكونة من عبارتين SELECT أو أكثر, يزيل الصفوف المكررة بين عبارات SELECT المختلفة.

**ملاحظة:** يجب أن يكون لكل عبارة SELECT داخل مشغل UNION نفس عدد الحقول في مجموعات النتائج مع أنواع بيانات مماثلة.

○ لنقوم بعرض الـID (قيم مميزة فقط) من كل من جدول "Customers" و "Orders":



**Result:**



**ملاحظة:** لا يمكن استخدام UNION لسرد كافة ID customers من الجدولين. استخدم UNION ALL لتحديد القيم المكررة.

#### ○ Union all:

هو مشابه جداً لـ Union, لكن تُرجع UNION فقط سجلاً فريداً ، بينما تُرجع UNION ALL جميع السجلات (بما في ذلك التكرارات).



**Result:**



#### ● Intersection:

يتم استخدام عامل التشغيل INTERSECT لإرجاع نتائج 2 أو أكثر من عبارات SELECT. ومع ذلك ، فإنه يقوم فقط بإرجاع الصفوف المحددة بواسطة كافة الاستعلامات أو مجموعات البيانات ويزيل INTERSECT التكرارات.

- نريد عرض ID الـ Customers الذين لديهم طلبات, باستخدام الـ Intersection؟



**Result:**



### الـ Intersect Vs. Inner join:

قد نرا تشابة كبير بينهم و قد تختلط لدينا الامور, لكن الإثنان مختلفان جدا. INNER JOIN هو عامل يتطابق بشكل عام مع مجموعة محدودة من الأعمدة ويمكن أن يُرجع صفراً من الصفوف أو المزيد من الصفوف من أي جدول. INTERSECT هو عامل تشغيل قائم على مجموعة يقارن الصفوف الكاملة بين مجموعتين ولا يمكنه أبداً إرجاع صفوف أكثر من الجدول الأصغر.

### • الـ Minus:

هو استعمال يستخدم عامل الطرح في SQL لطرح مجموعة نتائج واحدة من مجموعة نتائج أخرى لتقييم فرق مجموعة النتائج.

- نريد عرض ID الـ Customers الذين لم يكن لديهم اي طلبات, باستخدام الـ Minus ؟



**Result:**



دورة منصة سطر:

- المستوى المبتدئ

<https://sitr.codes/courses/FtkmhtJpQW/view>



• منصة سطر التعليمية [sitr.codes](https://sitr.codes)

- مستوى متوسط

<https://sitr.codes/courses/APjgdQqVWR/view>



• منصة سطر التعليمية [sitr.codes](https://sitr.codes)

- مستوى متقدم

<https://sitr.codes/courses/bOXiOFzkMv/view>



• منصة سطر التعليمية [sitr.codes](https://sitr.codes)