

# SQL 2

## التعامل مع البيانات ٢ / Data CRUD 2

\* تعيين Primary Key بعد إنشاء الجدول. (لا خلافه للأمر (القول الموجدة سابقاً)

```
1. USE company;  
2. ALTER TABLE Users  
3. ADD PRIMARY KEY (ID);
```

\* العامل LIKE :- نذكر معاً مع where للبحث عن قيمة معينة ) نذكر معاً :

يمثل حرف أو سلسلة من حروف متقدمة ← %  
يمثل حرف واحد ← \_

```
1. Use company;  
2. SELECT * FROM Employee  
3. WHERE Employee_name LIKE 'A%';
```

(القليل الذي سأبحث فيه)

| Employee_Id | Employee_Name | Employee_Salary | Employee_Bonus |
|-------------|---------------|-----------------|----------------|
| 3           | Ali           | 5500            | NULL           |
| 4           | Ahmad         | 5500            | NULL           |
| 5           | Abdullah      | 5500            | NULL           |

عرض جوهر الوظيفة التي تبدأ بـ A

A بحث

Query 1

1. Use company;  
 2. SELECT \* FROM Employee  
 3. WHERE Employee\_name LIKE '%D';

Result Grid

| Employee_Id | Employee_Name | Employee_Salary | Employee_Bonus |
|-------------|---------------|-----------------|----------------|
| 3           | saad          | 5600            | 2000           |
| 4           | Ahmad         | 5500            |                |

عرض جميع الموظفين أصحاب

الاسماء التي تنتهي ب D

Query 1

1. Use company;  
 2. SELECT \* FROM Employee  
 3. WHERE Employee\_name LIKE 'A\_\_';

Result Grid

| Employee_Id | Employee_Name | Employee_Salary | Employee_Bonus |
|-------------|---------------|-----------------|----------------|
| 3           | Ali           | 5500            |                |

عرض جميع الموظفين أصحاب

الاسماء التي تبدأ ب A ويليها حرفين

فقط

Query 1

1. Use company;  
 2. SELECT \* FROM Employee  
 3. WHERE Employee\_name LIKE 'A%';

Result Grid

| Employee_Id | Employee_Name | Employee_Salary | Employee_Bonus |
|-------------|---------------|-----------------|----------------|
| 3           | Ali           | 5500            |                |

عند وجود عدة مسموٰت يوحُّدُنـا من عرض البيانات المختلفة . بالاعادة كل عمود يابـي نـظر فيـه **Distinct** : \* الـكـلـة :

تحـلـرـ

```
1. USE Company;
2. SELECT DISTINCT Employee_name
3. FROM Employee;
```

Result Grid | Filter Rows: | Search: | Export: |

| Employee_name |
|---------------|
| Ahmad         |
| Ali           |
| Sarah         |
| Abdullah      |
| Samar         |
| Faisal        |
| Khalid        |
| Norah         |

\***الـكـلـة AS** فـي الـبـاولـ وـالـآخـدـة .. تـحـمـلـ الـبـلـدـلـ أـوـ الـقـدرـ فـقـطـ فـيـ عـلـيـةـ الرـضـ أـوـ لـاستـراـضـ وـلكـ الـاسـمـ «ـصـلـيـ»

لاـيـعـنـيـ

```
1. USE company;
2. SELECT Employee_name AS Ename
3. FROM Employee; ↓
```

الـقـدرـ الـمـلـكـ بـعـادـةـ تـبـيـهـ فـيـ عـلـيـةـ الرـضـ

Result Grid | Filter Rows: | Search: | Export: |

| Ename    |
|----------|
| Ahmad    |
| Ahmad    |
| Ali      |
| Sarah    |
| Abdullah |
| Samar    |
| Faisal   |
| Khalid   |

True  $\leftarrow 1$   
False  $\leftarrow 0$

SELECT :- Operator \* المسار \*

العمليات الرياضية

Arithmetic Operators

+

=

AND

\*

<

BETWEEN حل العامل يحدد نطاق التحازن

-

>

IN حل العامل يداري أكثر الترتيب لمحرك في الناشئة

/

<=

OR

>=

<>

Comparison Operators

عمليات المقارنة  
True or false

العمليات المنطقية  
Logical operators

\* نتاليات في جدول آخر :- إننا نجول بنتاليات في جدول سابق - أقدر أهد الأعده بعد SELECT وتقدير حروف باستخدام الأمر WHERE

```
Query 1
1. USE company;
2. CREATE TABLE Manager
3. SELECT * FROM Employee WHERE Employee_salary >= 6000;
```

## Aggregate functions نظرة على

\* مفهومها :- مجموعة من المدال الرجودة هي SQL والتي تطلب على عبود معين وترجمة في درجة بسيطة

general syntax:-

SELECT Aggregate - Functions (column - name)  
FROM Table - name  
WHERE conditions;  
عن المغيرة حتى اخوان  
شود له .

المدال :

- AVG() ①
- MAX() ②
- MIN() ③
- SUM() ④
- COUNT() ⑤

يقتضى تتم primary Key ← أعد صاعداً فرضه \* لضمان  
ان كل الفيقي لها القيمة موجودة لأن لا خبرت عدد وفيها قيمة  
هذا الصنف ويكون النتيجة تاتي

Ex:

SELECT COUNT (\*)

FROM Employees

WHERE Salary > 20000

## Scalar functions نظرية على

السؤال اي تأثير الماء على ورقة ما ← Numeric functions ①  
قيمة رقمية

| الدالة                 | الوصف                              |
|------------------------|------------------------------------|
| POW()                  | ترجع قيمة رقم مرفوع إلى أس رقم آخر |
| SELECT POW(3, 2); → 9  |                                    |
| MOD()                  | ترجع باقي القسمة للعددين           |
| SELECT MOD(10, 3); → 1 |                                    |
| ABS()                  | ترجع القيمة المطلقة للعدد          |
| SELECT ABS(-4); → 4    |                                    |
| DIV()                  | ترجع قسمة عددين                    |
| SELECT 10 DIV 5; → 2   |                                    |

ـ تذكرنا من التلبيب في الخرس ترجع احاجيم رقمية  
String functions ②  
او قيم نصية

| الوصف  | الدالة                   |
|--|--------------------------|
| هذه الدالة ترجع قيمة ASCII للمدخلات. وترجع 0 اذا كانت المدخلات فارغة   | <u>char</u> <b>ASCII</b> |
| هذه الدالة ترجع قيمة binary للمدخلات، والمدخلات تكون على صورة رقم (BIGINT). وترجع NULL إذا كانت المدخلات فارغة | <b>BIN</b>               |
| هذه الدالة ترجع النص الناتج من لصق نصين او اكثر من المدخلات.   | <b>CONCAT</b>            |
| هذه الدالة ترجع نص يمثل قيمة hexadecimal لنص أو عدد عشرى من المدخلات.  | <b>HEX</b>               |
| هذه الدالة ترجع حروف النص إلى lowercase  | <b>LOWER</b>             |
| هذه الدالة ترجع حروف النص إلى uppercase  | <b>UPPER</b>             |

Ex:-

SELECT ASCII('A'); → 65