استرجاع البيانات المخزنة من الجداول باستخدام جملة الاستعلام(SELECT)

تُعد هذه المهارة حجر الأساس والقوة المحركة لـ لغة الاستعلام عن البيانات(DQL) ، حيث يتمثل دورها الوظيفي في قراءة وسحب البيانات الرقمية والنصية المحفوظة مسبقاً داخل مستودعات الجداول لعرضها على شاشة المستخدم دون التعديل على الأصل أو المساس بالبنية التحتية للقواعد. وتكمن الأهمية التعليمية والتقنية لجملة SELECT في كونها النافذة الوحيدة التي تسمح للتطبيقات والبرامج باستكشاف محتويات خادم الويب، وتعتمد آلية عملها على تمرير أمر المسح البرمجي إلى محرك القواعد، ليقوم فوراً بفحص السجلات، وتجميع الحقول المطلوبة، وإعادتها في شكل جدول نتائج مرئي ومنظم بدقة عالية.

خطوات كتابة أمر استرجاع البيانات:(SELECT)

1. **استدعاء أداة الاستعلام الأساسية**: تبدأ الجملة البرمجية دائماً بكتابة الكلمة المحجوزة SELECT بأحرف كبيرة للإعلان عن بدء عملية القراءة والاسترجاع من خادم البيانات.
2. **تحديد الحقول المستهدفة**: ترك مسافة فارغة واحدة، ثم الإعلان عن الأعمدة المراد سحب بياناتها سواء بوضع علامة التعميم *أو كتابة أسماء أعمدة محددة.

```
SELECT * FROM students ;
```

3. **تحديد مصدر البيانات**: الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM بأحرف كبيرة لتهيئة المحرك لاستقبال اسم المستودع الرقمي.
4. **تسمية الجدول الفعلي**: ترك مسافة فارغة واحدة بعد أداة المصدر، ثم كتابة المسمى الدقيق والتنفيذي للجدول المراد سحب السجلات من داخله مثل students .
5. **إغلاق جملة الاستعلام**: إنهاء أمر القراءة بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار جملة الاستعلام وجاهزيتها للتنفيذ الفوري على السيرفر.

Select الحقول المراد استرجاعها

From اسم الجدول;

```
SELECT *
FROM students;
```

استعلام من جدول واحد لعرض كافة الأعمدة باستخدام الرمز * أو تحديد أعمدة معينة

تُعد هذه المهارة الأداة البرمجية المخصصة للتحكم في كمية وحجم البيانات المسترجعة من الجدول، حيث يتمثل دورها الوظيفي في التخيير بين مسارين: إما استخدام رمز التعميم النجمي * لسحب كافة الحقول والأعمدة المكونة للجدول دفعة واحدة، أو استبداله بكتابة أسماء أعمدة منتقاة بدقة. وتكمن الأهمية التعليمية والعملية لتطبيق هذه المهارة في تدريب الطالب على تجنب سحب البيانات غير الضرورية، وتعتمد آلية عملها على توجيه محرك القواعد لقراءة الأعمدة المحددة فقط في سطر الاستعلام وإهمال باقي حقول الجدول، مما يضمن سرعة الاستجابة وعرض النتائج المطلوبة فقط بشكل منظم ومحدد.

خطوات كتابة أمر الاستعلام الشامل أو المحدد:

1. **استدعاء أداة الاستعلام:** تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT بأحرف كبيرة لتهيئة الخادم لقراءة البيانات.
2. **اختيار نمط العرض:** ترك مسافة فارغة، ثم وضع رمز النجمة * لعرض كافة الأعمدة، أو كتابة أسماء أعمدة معينة مع الفصل بينها بفاصلة عادية , (مثل student_name, student_email).

```
SELECT student_name, student_email
FROM students ;
```

3. **تحديد أداة مصدر الجدول:** الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM بأحرف كبيرة لتحديد جدول القراءة.
4. **تحديد اسم الجدول الفعلي:** ترك مسافة فارغة واحدة، ثم كتابة المسمى الدقيق للجدول المستهدف بالاستعلام (مثل students).
5. **إغلاق جملة الاستعلام:** إنهاء السكربت بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار أمر القراءة وجاهزيته للتنفيذ الفوري على السيرفر.

الصيغة الأولى: لعرض كافة الأعمدة

SELECT * FROM اسم الجدول ;

```
SELECT * FROM students ;
```

الصيغة الثانية: لعرض أعمدة محددة مسبقاً

SELECT اسم العمود الأول, اسم العمود الثاني **FROM** اسم الجدول ;

```
SELECT student_phone, student_name FROM students ;
```

تصفية نتائج الاستعلام المسترجعة بناءً على شروط محددة باستخدام الجملة الشرطية(WHERE)

تُعد هذه المهارة الأداة البرمجية الحاسمة للتحكم في فرز وتصفية السجلات المسترجعة من الجداول، حيث يتمثل دورها الوظيفي في منع عرض كامل بيانات الجدول دفعة واحدة، واستخلاص الصفوف التي تنطبق عليها شروط ومعايير معينة فقط. وتكمن الأهمية التعليمية والعملية للجملة الشرطية WHERE في تمكين الطالب من بناء استعلامات ذكية ومحددة (مثل عرض بيانات الطلاب الناجحين فقط، أو سحب سجلات الطلاب التابعين لقسم معين)؛ وتعتمد آلية عملها على قيام محرك قواعد البيانات بمسح سجلات الجدول صفاً تلو الآخر، ومقارنة بيانات الحقل بالشرط المصاغ، فإذا كان السجل مطابقاً للشرط تم عرضه في جدول النتائج، وإذا كان مخالفاً يتم استبعاده تماماً وتصفيته.

خطوات كتابة الجملة الشرطية وتصفية النتائج:(WHERE)

1. **استدعاء أداة الاستعلام**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT بأحرف كبيرة لتفعيل مسار قراءة البيانات.

```
SELECT * FROM students
```

2. **تحديد حقول العرض**: ترك مسافة فارغة واحدة، ثم الإعلان عن الأعمدة المراد إظهارها في النتائج مثل وضع علامة *لعرض كافة الأعمدة.
3. **تحديد جدول المصدر**: الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM متبوعة مباشرة باسم الجدول المستهدف بالبحث مثل students.

4. **استدعاء أداة التصفية الشرطية**: الانتقال لسطر جديد وكتابة العبارة المحجوزة WHERE بأحرف كبيرة لإعلام المحرك ببدء فرز السجلات.
5. **صياغة الشرط الحاكم وإغلاق الأمر**: كتابة المعادلة الشرطية مباشرة بعد الأداة (مثل student_id = 1)، ثم إنهاء السكربت بالفاصلة المنقوطة ; لتنفيذه على السيرفر.

```
WHERE student_id = 1 ;
```

اسم العمود المعامل المنطقي WHERE اسم الجدول FROM اسم العمود الأول, اسم العمود الثاني SELECT القيمة المستهدفة ;

```
SELECT * FROM students WHERE student_id = 1 ;
```

دمج مخرجات حقلين نصيين أو أكثر في عمود واحد مسترجع باستخدام دالة الدمج(CONCAT)

تُعد هذه المهارة الأداة البرمجية المخصصة لمعالجة النصوص وتنسيق مخرجات الاستعلام، حيث يتمثل دورها الوظيفي في دمج محتويات حقلين نصيين أو أكثر (أو نصوص ثابتة) وإظهارها مجتمعة داخل عمود واحد في جدول النتائج المسترجعة. وتكمن الأهمية التعليمية والعملية لدالة الدمج في توفير مخرجات برمجية جاهزة للقراءة ومترابطة (مثل دمج الاسم الأول مع الاسم الأخير لتكوين الاسم الكامل للطالب، أو دمج كود الطالب مع اسمه)؛ وتعتمد آلية عملها على قيام المحرك بتركيب السلاسل النصية بداخل الاستعلام وعرضها ككتلة واحدة مرئية، دون التغيير في البيانات الأصلية المخزنة داخل الجدول.

خطوات كتابة أمر دمج الحقول النصية:(CONCAT)

1. **استدعاء أداة الاستعلام**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT بأحرف كبيرة لتهيئة الخادم لبدء القراءة.
2. **استدعاء دالة الدمج النصي**: ترك مسافة فارغة واحدة، ثم كتابة الكلمة المحجوزة CONCAT متبوعة مباشرة بفتح قوسين هلاليين ( ) لحصر الحقول المراد دمجها.

3. **تحديد الحقول والمواصلات النصية** :كتابة أسماء الأعمدة المراد دمجها داخل الأقواس مع الفصل بينها بفاصلة عادية) ,ويمكن وضع مسافة نصية تفصل بينهما مثل ' ', .
4. **تحديد جدول المصدر** :الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM متبوعة بالاسم الفعلي للجدول المراد سحب السجلات من داخله مثل students.
5. **إغلاق جملة الاستعلام** :إنهاء السكربت بوضع الفاصلة المنقوطة ;للإشارة إلى نهاية مسار الأمر الحالي وجاهزيته للتنفيذ الفوري على السيرفر.

```
SELECT CONCAT(first_name, ' ', last_name)
FROM students ;
```

SELECT CONCAT(اسم العمود الأول, ' النص الفاصل ', اسم العمود الثاني) **FROM** اسم الجدول ;

```
SELECT CONCAT(first_name, ' ', last_name) FROM students ;
```

(اختياري) عادة ما يتم إلحاق دالة الدمج بكلمة ASلإعطاء اسم مستعار للعمود الجديد المدمج مثل AS full_name حتى يظهر بشكل مرتب ومقروء في نتائج البحث بدلاً من ظهور معادلة الدمج كاسم للعمود.

```
SELECT CONCAT(first_name, ' ', last_name) AS full_name
FROM students ;
```

تسمية الحقول المسترجعة بأسماء مستعارة مؤقتة لتسهيل قراءتها في النتائج باستخدام الأمر(AS / Alias)

تُعد هذه المهارة الأداة البرمجية المخصصة لتنظيم وتجميل مظهر تقارير البيانات المسترجعة، حيث يتمثل دورها الوظيفي في منح الأعمدة أو الجداول أسماءً بديلة ومؤقتة (Alias) تظهر فقط في شاشة النتائج دون أن يؤدي ذلك إلى تغيير الأسماء الحقيقية المخزنة في بنية الجدول الأصلية. وتكمن الأهمية التعليمية والعملية لتطبيق الأسماء المستعارة في تبسيط قراءة المخرجات للمستخدم النهائي، لا سيما عند التعامل مع دالات الدمج النصي (مثل دالة CONCAT) أو العمليات الحسابية المعقدة التي تنتج أعمدة بدون مسمى

واضح؛ وتعتمد آلية عملها على تمرير الكلمة المحجوزة ASمتبوعة بالاسم الجديد، ليقوم المحرك فوراً باستبدال عنوان الحقل في جدول المخرجات المرئي.

خطوات كتابة أمر تعيين الأسماء المستعارة:(AS)

1. **استدعاء أداة الاستعلام** :تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT بأحرف كبيرة لبدء تفعيل مسار القراءة.
2. **تحديد الحقل واستدعاء أداة التسمية** :كتابة اسم العمود الفعلي المراد سحبه، متبوعاً بمسافة فارغة واحدة، ثم كتابة الكلمة المفتاحية AS بأحرف كبيرة.
3. **تحديد الاسم البديل المستهدف** :ترك مسافة فارغة وكتابة المسمى المؤقت المراد ظهوره في جدول النتائج مثل Student Name
4. **تحديد جدول المصدر الفعلي** :الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM متبوعة بالاسم الصريح للجدول مثل students
5. **إغلاق جملة الاستعلام وعرض النتائج** :إنهاء السكربت بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار الأمر وجاهزيته للتنفيذ الفوري على السيرفر.

```
SELECT student_name AS 'Student Name'
FROM students ;
```

SELECT اسم العمود الأصلي **AS** الاسم المستعار الجديد **FROM** اسم الجدول ;

```
SELECT student_name AS Student_Full_Name FROM students ;
```

ملاحظة إضافية: يمكن تطبيق هذا الأمر على أكثر من عمود في نفس الاستعلام عن طريق الفصل بينها بفاصلة عادية، هكذا:

```
SELECT student_name AS 'Student Name', age AS 'Student Age'
FROM students ;
```

ترتيب سجلات النتائج تصاعدياً أو تنازلياً ويحدد عدد الصفوف المسترجعة باستخدام المعاملات (ORDER LIMIT / BY)

تُعد هذه المهارة الأداة البرمجية المخصصة للتحكم في تنظيم كمية ومظهر البيانات المسترجعة، حيث يتمثل دورها الوظيفي في شقين: الأول هو إعادة ترتيب صفوف النتائج تصاعدياً أو تنازلياً بناءً على قيم حقل معين (مثل ترتيب الطلاب أبجدياً أو حسب درجاتهم)، والثاني هو حصر وتقييد عدد السجلات المعروضة على الشاشة برقم محدد. وتكمن الأهمية التعليمية والعملية لتطبيق هذه المهارة في تمكين الطالب من عرض التقارير الإحصائية الهامة (مثل عرض أسماء الطلاب الثلاثة الأوائل فقط في النتيجة)؛ وتعتمد آلية عملها على قيام المحرك بفرز السجلات في الذاكرة أولاً بناءً على معامل الترتيب، ثم قطع وعرض العدد المطلوب فقط وإهمال باقي الصفوف لتسريع الاستجابة.

خطوات كتابة أمر ترتيب وتقييد السجلات(ORDER BY / LIMIT)

1. **استدعاء أداة الاستعلام وعناصر العرض** :تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة بالحقول المراد عرضها ثم تحديد الجدول عبر العبارة FROM اسم الجدول.

```
SELECT student_name, grade_score
FROM students
```

2. **استدعاء أداة التترتيب** :الانتقال لسطر جديد وكتابة العبارة المحجوزة ORDER BY بأحرف كبيرة متبوعة مباشرة باسم العمود المراد الترتيب بناءً عليه مثل grade_score

```
ORDER BY grade_score DESC
```

3. **تحديد اتجاه الفرز** :كتابة كلمة ASC بعد اسم الحقل لجعل الترتيب تصاعدياً (من الأصغر للأكبر)، أو كتابة كلمة DESC لجعله تنازلياً (من الأكبر للأصغر).
4. **تحديد حجم الصفوف المسترجعة** :الانتقال لسطر جديد وكتابة الكلمة المفتاحية LIMIT بأحرف كبيرة لتقييد العدد، يليه مباشرة الرقم المطلوب مثل 3 .

```
LIMIT 3 ;
```

5. **إغلاق جملة الاستعلام**: إنهاء السكربت بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار الأمر الحالي وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT اسم العمود **FROM** اسم الجدول **ORDER BY** اسم العمود اتجاه الترتيب **LIMIT** عدد الصفوف ;

```
SELECT student_name, grade_score FROM student_grades ORDER BY
grade_score DESC LIMIT 3 ;
```

إنشاء الفهارس Indexes على الأعمدة الحيوية لتسريع عمليات البحث وتقليل جهد معالج السيرفر

تُعد هذه المهارة الأداة البرمجية المخصصة لرفع كفاءة وسرعة استجابة قاعدة البيانات عند البحث، حيث يتمثل دورها الوظيفي في بناء كشاف فهرسي مستقل (Index) لحقول معينة يكثر الاستعلام عنها (مثل أسماء الطلاب أو الأرقام القومية). وتكمن الأهمية التعليمية والتقنية لتطبيق الفهارس في ترشيد استهلاك موارد الخادم؛ فعوضاً عن قيام معالج السيرفر بفحص ملايين السجلات صفاً تلو الآخر، ينتقل مباشرة إلى موقع السجل عبر الفهرس؛ وتعتمد آلية عملها على تسريع عملية فرز واسترجاع البيانات بأقل جهد وأعلى سرعة ممكنة، مما يمنع بطء أو توقف النظام أثناء الاستعلامات الضخمة.

خطوات كتابة أمر إنشاء الفهرس:(CREATE INDEX)

1. **استدعاء أداة التأسيس الفهرسي**: تبدأ الجملة البرمجية بكتابة العبارة المحجوزة CREATE INDEX بأحرف كبيرة للإعلان عن بناء كائن فهرسي جديد.

```
CREATE INDEX idx_student_name
ON students (student_name) ;
```

2. **تسمية الفهرس المستجد**: ترك مسافة فارغة واحدة، ثم كتابة اسم رمزي خاص بالفهرس لتسهيل تتبعه وإدارته لاحقاً مثل idx_student_name

3. **تحديد أداة ربط المستودع** :ترك مسافة فارغة، ثم كتابة الكلمة المفتاحية ON بأحرف كبيرة لتوجيه المحرك نحو الجدول المستهدف.
4. **تحديد الجدول والحقل الحيوي** :كتابة الاسم الفعلي للجدول مثل students متبوعة مباشرة بفتح قوسين هلاليين ( ) يكتب بينهما اسم العمود المراد فرزه وفهرسته مثل student_name
5. **إغلاق جملة الاستعلام الهيكلية** :إنهاء السكربت بوضع الفاصلة المنقوطة ;للإشارة إلى نهاية مسار الأمر الهيكلي وجاهزيته للتنفيذ الفوري على السيرفر.

CREATE INDEX اسم الفهرس **ON** (اسم العمود) اسم الجدول ;

```
CREATE INDEX idx_student_name ON students (student_name) ;
```

استخراج قيمة تبدأ بحرف معين باستخدام المعامل (LIKE) والرمز المئوي(%)

تُعد هذه المهارة الأداة البرمجية المخصصة لتصنيف وتصفية النصوص بناءً على سابقتها اللفظية، حيث يتمثل دورها الوظيفي في استرجاع السجلات التي تبدأ قيمها النصية بحرف أو مقطع محدد فقط وإهمال بقية السجلات. وتكمن الأهمية التعليمية والعملية لتطبيق هذا النمط في تمكين الطالب من بناء فلاتر البحث الأبجدية (مثل عرض أسماء الطلاب الذين تبدأ أسماؤهم بحرف "أ" أو "م")؛ وتعتمد آلية عملها على صياغة الحرف المستهدف متبوعاً مباشرة بالرمز المئوي) %مثل 'أ'(%، ليفهم المحرك أن الرمز يعوض أي عدد من الحروف اللاحقة بشرط أن يكون الحرف الأول ثابتاً ومطابقاً.

خطوات كتابة أمر استخراج قيمة تبدأ بحرف معين:

1. **استدعاء أداة الاستعلام وعناصر العرض** :تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة بالحقول المراد عرضها ثم تحديد الجدول عبر العبارة FROM اسم الجدول.

```
SELECT student_name
FROM students
```

2. **استدعاء أداة التصفية وتحديد الحقل المستهدف**: الانتقال لسطر جديد وكتابة العبارة المحجوزة WHERE بأحرف كبيرة متبوعة مباشرة باسم العمود النصي مثل student name.
3. **تضمين معامل المطابقة النصية**: ترك مسافة فارغة واحدة، ثم كتابة الكلمة المفتاحية LIKE بأحرف كبيرة لتوجيه السيرفر نحو البحث الجزئي.
4. **صياغة نمط البداية الحاكم**: ترك مسافة وكتابة الحرف المستهدف محاطاً بعلامات تنصيص مفردة مع وضع الرمز المئوي %**بعد الحرف مباشرة** مثل 'أ%' .

```
WHERE student_name LIKE 'أ%' ;
```

5. **إغلاق جملة الاستعلام البرمجية**: إنهاء السكربت بالكامل بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار الأمر وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT اسم العمود **FROM** اسم الجدول **WHERE** اسم العمود **LIKE** 'الحرف المستهدف%' ;

```
SELECT * FROM students WHERE student_name LIKE 'أ%' ;
```

استخرج قيمة تنتهي بحرف معين باستخدام المعامل (LIKE) والرمز المئوي(%)

تُعد هذه المهارة الأداة البرمجية المخصصة لفرز السجلات بناءً على خواتيمها النصية، حيث يتمثل دورها الوظيفي في استخلاص الصفوف التي تنتهي قيمها النصية بحرف أو مقطع محدد وإهمال ما دونها. وتكمن الأهمية التعليمية والعملية لتطبيق هذا النمط في تدريب الطالب على فرز البيانات وتحليلها وفق نهاياتها (مثل استخراج البريد الإلكتروني الذي ينتهي بامتداد معين، أو البحث عن الأسماء التي تنتهي بحرف معين كحفل التخرج)؛ وتعتمد آلية عملها على وضع الرمز المئوي % في البداية متبوعاً بالحرف المستهدف (مثل '%م')، ليفهم المحرك أن الرموز السابقة متغيرة ولكن الحرف الأخير يجب أن يكون ثابتاً ومطابقاً.

خطوات كتابة أمر استخراج قيمة تنتهي بحرف معين:

1. **استدعاء أداة الاستعلام وعناصر العرض**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة بالحقول المراد عرضها ثم تحديد الجدول عبر العبارة FROM اسم الجدول.

```
SELECT student_name
FROM students
```

2. **استدعاء أداة التصفية وتحديد الحقل المستهدف**: الانتقال لسطر جديد وكتابة العبارة المحجوزة WHERE بأحرف كبيرة متبوعة مباشرة باسم العمود النصي مثل student_name
3. **تضمين معامل المطابقة النصية**: ترك مسافة فارغة واحدة، ثم كتابة الكلمة المفتاحية LIKE بأحرف كبيرة لتوجيه السيرفر نحو البحث الجزئي.
4. **صياغة نمط النهاية الحاكم**: ترك مسافة وكتابة الحرف المستهدف محاطاً بعلامات تنصيص مفردة مع وضع الرمز المئوي %**قبل الحرف مباشرة** مثل %'م' .

```
WHERE student_name LIKE '%م' ;
```

5. **إغلاق جملة الاستعلام البرمجية**: إنهاء السكربت بالكامل بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار الأمر وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT اسم العمود **FROM** اسم الجدول **WHERE** اسم العمود **LIKE** '%الحرف المستهدف' ;

```
SELECT * FROM students WHERE student_name LIKE '%م' ;
```

استخراج قيمة تحتوي على حرف معين باستخدام المعامل (LIKE) والرمز المئوي(%)

تُعد هذه المهارة الأداة البرمجية الشاملة والأكثر مرونة للبحث النصي الحر، حيث يتمثل دورها الوظيفي في استرجاع كافة السجلات التي تتضمن الحرف أو المقطع المستهدف في أي موقع داخل النص (سواء في البداية، الوسط، أو النهاية). وتكمن الأهمية التعليمية والعملية لتطبيق نمط الاحتواء في بناء محركات البحث الذكية الشبيهة بأنظمة البحث الحقيقية (مثل تفويض الطالب لكتابة جزء من اسم الطالب أو جزء من عنوانه

واستدعاء النتائج فوراً)؛ وتعتمد آلية عملها على إحاطة الحرف المستهدف برمزين مئويين من الجهتين مثل %'احمد'%، ليعلم المحرك أن الشرط الوحيد للقبول هو وجود هذا اللفظ بداخل النص بغض النظر عما يسبقه أو يلحقه.

خطوات كتابة أمر استخراج قيمة تحتوي على حرف معين:

1. **استدعاء أداة الاستعلام وعناصر العرض**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة بالحقول المراد عرضها ثم تحديد الجدول عبر العبارة FROM اسم الجدول.
2. **استدعاء أداة التصفية وتحديد الحقل المستهدف**: الانتقال لسطر جديد وكتابة العبارة المحجوزة WHERE بأحرف كبيرة متبوعة مباشرة باسم العمود النصي مثل student_name
3. **تضمين معامل المطابقة النصية**: ترك مسافة فارغة واحدة، ثم كتابة الكلمة المفتاحية LIKE بأحرف كبيرة لتوجيه السيرفر نحو البحث الجزئي.
4. **صياغة نمط الاحتواء الحاكم**: ترك مسافة وكتابة الحرف المستهدف محاطاً بعلامات تنصيص مفردة مع وضع الرمز المئوي **%من الجهتين قبل وبعد الحرف** مثل %'ع'%

```
SELECT student_name
FROM students
WHERE student_name LIKE '%ع%' ;
```

5. **إغلاق جملة الاستعلام البرمجية**: إنهاء السكربت بالكامل بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار الأمر وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT اسم العمود **FROM** اسم الجدول **WHERE** اسم العمود **LIKE** '%الحرف المستهدف%' ;

SELECT * FROM students **WHERE** student_name **LIKE '%ع%'** ;

تطبيق شروطاً متعددة ومركبة داخل استعلام التصفية باستخدام المعامل المنطقي الإلزامي(AND)

تُعد هذه المهارة الأداة البرمجية المتقدمة لبناء استعلامات ذكية وشديدة الدقة، حيث يتمثل دورها الوظيفي في دمج أكثر من شرط معاً داخل جملة التصفية لفرز السجلات بربط إلزامي. وتكمن الأهمية التعليمية والعملية لتطبيق المعامل المنطقي AND في تمكين الطالب من صياغة قيود مركبة؛ حيث يُستخدم هذا المعامل عندما يشترط النظام **تحقق جميع الشروط المصاغة معاً في نفس الوقت** لإظهار السجل (مثل عرض الطلاب الناجحين والتابعين لقسم التكنولوجيا معاً، فإذا تخلف شرط واحد يستبعد السجل)؛ وتعتمد آلية عملها على قيام المحرك بتقييم الشروط مجتمعة لكل صف، وبناءً على موافقتها الكاملة للشرطين يتم قبول السجل.

خطوات كتابة استعلام التصفية بالشرط المركب:(AND)

1. **استدعاء أداة الاستعلام وعناصر العرض**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة بالحقول المراد عرضها ثم تحديد الجدول عبر العبارة FROM اسم الجدول.

```
SELECT student_name
FROM students
WHERE department = 'Tech' AND grade_score >= 85 ;
```

2. **استدعاء أداة التصفية وتحديد الشرط الأول**: كتابة العبارة المحجوزة WHERE بأحرف كبيرة متبوعة مباشرة بالشرط الأساسي الأول مثل department = 'Tech'
3. **دمج المعامل المنطقي الحاكم**: ترك مسافة فارغة واحدة، ثم كتابة المعامل المطلوب بأحرف كبيرة AND للربط الإلزامي المشترك.
4. **صياغة الشرط الثاني والتكميلي**: ترك مسافة فارغة بعد المعامل، ثم كتابة الشرط الثاني المراد فحص السجلات بناءً عليه مثل grade_score >= 85
5. **إغلاق جملة الاستعلام المركبة**: إنهاء السكربت بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار الأمر الحالي وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT اسم العمود **FROM** اسم الجدول **WHERE** الشرط الأول **AND** الشرط الثاني ;

```
SELECT * FROM students WHERE department = 'Tech' AND grade_score >= 85 ;
```

تطبيق شروطاً متعددة ومركبة داخل استعلام التصفية باستخدام المعامل المنطقي الاختياري(OR)

تُعد هذه المهارة الأداة البرمجية المرنة لتوسيع نطاق البحث والتصفية داخل الجداول، حيث يتمثل دورها الوظيفي في دمج عدة شروط معاً داخل جملة التصفية بربط اختياري ميسر. وتكمن الأهمية التعليمية والعملية لتطبيق المعامل المنطقي OR في تمكين الطالب من صياغة قيود بديلة؛ حيث يُستخدم هذا المعامل عندما **يكفي تحقق شرط واحد على الأقل من الشروط المصاغة** لعرض السجل (مثل عرض الطلاب التابعين لقسم التكنولوجيا أو الطلاب الذين يسكنون في محافظة المنيا، فيكفي أن ينطبق أي شرط منهما ليظهر الطالب في النتائج)؛ وتعتمد آلية عملها على تمرير الصفوف التي توافق أي بند من البنود المذكورة.

خطوات كتابة استعلام التصفية بالشرط المركب:(OR)

1. **استدعاء أداة الاستعلام وعناصر العرض**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة بالحقول المراد عرضها ثم تحديد الجدول عبر العبارة FROM اسم الجدول.

```
SELECT student_name
FROM students
WHERE department = 'Tech' OR city = 'Minia' ;
```

2. **استدعاء أداة التصفية وتحديد الشرط الأول**: كتابة العبارة المحجوزة WHERE بأحرف كبيرة متبوعة مباشرة بالشرط البديل الأول مثل department = 'Tech'
3. **دمج المعامل المنطقي الحاكم**: ترك مسافة فارغة واحدة، ثم كتابة المعامل المطلوب بأحرف كبيرة OR للربط الاختياري التبادلي.
4. **صياغة الشرط الثاني البديل**: ترك مسافة فارغة بعد المعامل، ثم كتابة الشرط الثاني المراد فحص السجلات بناءً عليه مثل city = 'Minia' .
5. **إغلاق جملة الاستعلام المركبة**: إنهاء السكربت بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار الأمر الحالي وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT اسم العمود **FROM** اسم الجدول **WHERE** الشرط الأول **OR** الشرط الثاني ;

```
SELECT * FROM students WHERE department = 'Tech' OR city = 'Minia' ;
```

توظيف دالة التجميع (SUM) لحساب المجموع الإجمالي للقيم

تُعد هذه المهارة الأداة البرمجية المخصصة لإيجاد العمليات الحسابية التراكمية، حيث يتمثل دورها الوظيفي في جمع كافة القيم الرقمية المخزنة داخل حقل معين لجميع سجلات الجدول وإظهار الناتج الإجمالي في قيمة واحدة مسترجعة. وتكمن الأهمية التعليمية والعملية لتطبيق دالة المجموع SUMفي تمكين الطالب من استخراج التقارير المالية والعددية الضخمة (مثل حساب إجمالي المصاريف الدراسية التي تم تحصيلها، أو مجموع درجات الساعات المعتمدة للطلاب)؛ وتعتمد آلية عملها على قيام المحرك بالمرور على العمود الرقمي المحدد صفاً تلو الآخر وتجميع قيمه تصاعدياً حتى نهاية السجلات المعنية.

خطوات كتابة واستخدام دالة التجميع:(SUM)

1. **استدعاء أداة الاستعلام القياسية**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT بأحرف كبيرة لبدء مسار المعالجة والعرض الإحصائي.

```
SELECT SUM(payment_amount)
FROM student_payments ;
```

2. **تحديد الدالة التجميعية المستهدفة**: ترك مسافة فارغة واحدة، ثم كتابة اسم الدالة SUM بأحرف كبيرة لتهيئة المحرك لعملية الجمع الحسابي.
3. **تحديد الحقل الخاضع للجمع الحسابي**: فتح قوسين هلاليين ( ) مباشرة بعد اسم الدالة، وكتابة اسم العمود الرقمي المراد جمع قيم حقوله بداخلها مثل payment amount
4. **تحديد جدول المصدر الفعلي**: الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM متبوعة مباشرة بالاسم الصريح للجدول مثل student payments.
5. **إغلاق جملة الاستعلام الإحصائية**: إنهاء السكربت بالكامل بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار الأمر الحالي وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT SUM(اسم العمود) **FROM** اسم الجدول ;

SELECT SUM(payment_amount) **FROM** student_payments ;

لتجميل شكل المخرجات وتسهيل قراءتها، يُفضل دائماً استخدام أداة التسمية المستعارة (AS) لإعطاء اسم واضح لعمود المجموع، هكذا:

```
SELECT SUM(payment_amount) AS 'Total Payments'
FROM student_payments ;
```

توظيف دالة عد القيم (COUNT) لحساب العدد الإجمالي للسجلات

تُعد هذه المهارة الأداة البرمجية المخصصة لإحصاء وحصر حجم البيانات، حيث يتمثل دورها الوظيفي في حساب العدد الإجمالي للصفوف أو السجلات المخزنة داخل الجدول والتي تحتوي على بيانات فعلية، وإظهار هذا العدد في مخرج واحد. وتكمن الأهمية التعليمية والعملية لتطبيق دالة العد COUNT في تدريب الطالب على استخراج المؤشرات العددية والكمية الحيوية للأنظمة (مثل حساب عدد الطلاب الإجمالي المسجلين في المقرر، أو عدد الشعب المتاحة)؛ وتعتمد آلية عملها على قيام المحرك بمسح معرفات السجلات وحساب قوامها العددي الإجمالي مباشرة دون النظر إلى محتواها الرقمي.

خطوات كتابة واستخدام دالة عد القيم:(COUNT)

1. **استدعاء أداة الاستعلام القياسية** :تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT بأحرف كبيرة لبدء مسار المعالجة والعرض الإحصائي.

```
SELECT COUNT(student_id)
FROM students ;
```

2. **تحديد دالة العد المطلوبة** :ترك مسافة فارغة واحدة، ثم كتابة اسم الدالة COUNT بأحرف كبيرة لتهيئة المحرك لبدء عملية الإحصاء العددي.
3. **تحديد الحقل الخاضع للعد** :فتح قوسين هلاليين ( ) مباشرة بعد اسم الدالة، وكتابة اسم العمود المراد عد صفوفه مثل student id أو وضع النجمة لعد كامل السجلات.
4. **تحديد جدول المصدر الفعلي** :الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM متبوعة مباشرة بالاسم الصريح للجدول مثل students .

5. **إغلاق جملة الاستعلام الإحصائية**: إنهاء السكربت بالكامل بوضع الفاصلة المنقوطة ;للإشارة إلى نهاية مسار الأمر الحالي وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT COUNT(اسم العمود) **FROM** اسم الجدول ;

```
SELECT COUNT(student_id) FROM students ;
```

لعد جميع السجلات في الجدول بشكل عام دون التقيد بعمود معين، وإعطاء النتيجة اسماً مستعاراً يسهل قراءته، يمكنك كتابة الكود بهذه الطريقة:

```
SELECT COUNT(*) AS 'Total Students'
FROM students ;
```

توظيف دالة المتوسط (AVG) لحساب المتوسط الحسابي للقيم

تُعد هذه المهارة الأداة البرمجية المخصصة لاستخراج المعدلات الرياضية ومؤشرات الأداء، حيث يتمثل دورها الوظيفي في حساب الوسط أو المتوسط الحسابي لقيم حقل رقمي معين عبر جمع القيم الإجمالي ثم قسمتها تلقائياً على عدد السجلات. وتكمن الأهمية التعليمية والعملية لتطبيق دالة المعدل AVG في تمكين الطالب من إنتاج تقارير تحليلية ومقارنات علمية (مثل حساب متوسط درجات الطلاب في اختبار مادة قواعد البيانات، أو متوسط أعمار الطلاب المستجدين)؛ وتعتمد آلية عملها على دمج معالجة الجمع والعد معاً ديناميكياً لتوليد القيمة الوسطية الحاكمة.

خطوات كتابة واستخدام دالة المتوسط الحسابي:(AVG)

1. **استدعاء أداة الاستعلام القياسية**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT بأحرف كبيرة لبدء مسار المعالجة والعرض الإحصائي.

```
SELECT AVG(exam_score)
FROM student_exams ;
```

2. **تحديد دالة المتوسط المطلوبة**: ترك مسافة فارغة واحدة، ثم كتابة مسمى الدالة AVG بأحرف كبيرة لتوجيه السيرفر نحو الحساب المعدلي.
3. **تحديد الحقل الخاضع لحساب المعدل**: فتح قوسين هلاليين ( ) مباشرة بعد اسم الدالة، وكتابة اسم العمود الرقمي المراد حساب متوسطه مثل exam score
4. **تحديد جدول المصدر الفعلي**: الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM متبوعة مباشرة بالاسم الصريح للجدول مثل student exams
5. **إغلاق جملة الاستعلام الإحصائية**: إنهاء السكربت بالكامل بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار الأمر الحالي وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT AVG(اسم العمود) **FROM** اسم الجدول ;

```
SELECT AVG(exam_score) FROM student_exams ;
```

لجعل النتائج أكثر وضوحاً عند القراءة، يُفضل استخدام أداة التسمية المستعارة (AS) لمنح عمود النتيجة اسماً دلالياً، هكذا:

```
SELECT AVG(exam_score) AS 'Average Score'
FROM student_exams ;
```

توظيف دالة القيمة العظمى (MAX) لاستخراج أعلى قيمة رقمية

تُعد هذه المهارة الأداة البرمجية المخصصة لتحديد سقف البيانات والقمم الرقمية، حيث يتمثل دورها الوظيفي في فحص سجلات حقل معين بالكامل واستخلاص القيمة الأكبر أو الأعلى عدداً من بينها وإهمال ما دونها. وتكمن الأهمية التعليمية والعملية لتطبيق دالة النهاية العظمى MAX في رصد التميز والحدود القصوى داخل النظام (مثل استخراج أعلى درجة حصل عليها طالب في الامتحان، أو رصد الحد الأقصى للميزانيات المرصودة للأنشطة)؛ وتعتمد آلية عملها على فرز ومقارنة تصاعدية سريعة داخل الذاكرة لعزل القيمة الكبرى وعرضها منفردة.

خطوات كتابة واستخدام دالة القيمة العظمى:(MAX)

1. **استدعاء أداة الاستعلام القياسية** :تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT بأحرف كبيرة لبدء مسار المعالجة والعرض الإحصائي.

```
SELECT MAX(total_marks)
FROM results ;
```

2. **تحديد دالة النهاية العظمى** :ترك مسافة فارغة واحدة، ثم كتابة اسم الدالة MAX بأحرف كبيرة لإعلام السيرفر ببدء تصفية الماكسيمم.
3. **تحديد الحقل الخاضع للفرز الأقصى** :فتح قوسين هلاليين ( ) مباشرة بعد اسم الدالة، وكتابة اسم العمود الرقمي المستهدف بالمقارنة مثل total marks.
4. **تحديد جدول المصدر الفعلي** :الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM متبوعة مباشرة بالاسم الصريح للجدول مثل results .
5. **إغلاق جملة الاستعلام الإحصائية** :إنهاء السكربت بالكامل بوضع الفاصلة المنقوطة ;للإشارة إلى نهاية مسار الأمر الحالي وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT MAX(اسم العمود) **FROM** اسم الجدول ;

SELECT MAX(total_marks) **FROM** results ;

لجعل مخرجات الاستعلام أكثر دلالة واحترافية عند العرض، يُفضل استخدام أداة التسمية المستعارة (AS) لمنح عمود النتيجة اسماً واضحاً، هكذا:

```
SELECT MAX(total_marks) AS 'Highest Mark'
FROM results ;
```

توظيف دالة القيمة الصغرى (MIN) لاستخراج أقل قيمة رقمية

تُعد هذه المهارة الأداة البرمجية المخصصة لتحديد قاع البيانات والحدود الدنيا للقيم، حيث يتمثل دورها الوظيفي في فحص كافة المدخلات داخل حقل رقمي محدد واستخلاص القيمة الأصغر أو الأدنى عدداً

وعرضها في جدول المخرجات. وتكمن الأهمية التعليمية والعملية لتطبيق دالة النهاية الصغرى MINفي رصد الضعف أو التكلفة الأقل في بيئة التطوير (مثل استخراج أقل درجة حصل عليها طالب لتحديد فئات الدعم الأكاديمي، أو تحديد أدنى سعر لمنتج)؛ وتعتمد آلية عملها على إجراء عملية تصفية تنازلية للقيم داخل محرك القواعد لعزل القيمة الصغرى بدقة.

خطوات كتابة واستخدام دالة القيمة الصغرى:(MIN)

1. **استدعاء أداة الاستعلام القياسية** :تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT بأحرف كبيرة لبدء مسار المعالجة والعرض الإحصائي.
2. **تحديد دالة النهاية الصغرى** :ترك مسافة فارغة واحدة، ثم كتابة اسم الدالة MIN بأحرف كبيرة لتوجيه السيرفر نحو فرز المينيمم.

```
SELECT MIN(total_marks)
FROM results ;
```

3. **تحديد الحقل الخاضع للفرز الأدنى** :فتح قوسين هلاليين ( ) مباشرة بعد اسم الدالة، وكتابة اسم العمود الرقمي المراد جلب قيمته الصغرى مثل total_marks.
4. **تحديد جدول المصدر الفعلي** :الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM متبوعة مباشرة بالاسم الصريح للجدول مثل results .
5. **إغلاق جملة الاستعلام الإحصائية** :إنهاء السكربت بالكامل بوضع الفاصلة المنقوطة ;للإشارة إلى نهاية مسار الأمر الحالي وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT MIN(اسم العمود) **FROM** اسم الجدول ;

SELECT MIN(total_marks) FROM results ;

يُفضل استخدام أداة التسمية المستعارة (AS) لمنح عمود النتيجة اسماً دلالياً يسهل قراءته وفهمه في جدول النتائج النهائية، هكذا:

```
SELECT MIN(total_marks) AS 'Lowest Mark'
FROM results ;
```

تجميع السجلات المسترجعة بناءً على حقول معينة باستخدام جملة التجميع(GROUP BY)

تُعد هذه المهارة الأداة البرمجية الأساسية لتصنيف البيانات وبناء التقارير الفئوية، حيث يتمثل دورها الوظيفي في تلخيص السجلات المتشابهة ودمجها داخل حزم أو مجموعات فرعية مشتركة بناءً على قيم عمود محدد. وتكمن الأهمية التعليمية والعملية لتطبيق جملة GROUP BY في تمكين الطالب من الانتقال من مستوى استعراض الأفراد إلى مستوى استعراض المجموعات الإحصائية (مثل تجميع الطلاب حسب أقسامهم الأكاديمية لمعرفة قوام كل قسم، أو تجميع المدفوعات حسب سنة التحصيل)؛ وتعتمد آلية عملها على فرز صفوف الجدول وتجميع الحقول المتطابقة معاً بالتزامن مع وظيفة الدوال التجميعية لإخراج صف واحد يمثل كل فئة.

خطوات كتابة أمر تجميع السجلات فئوياً:(GROUP BY)

1. **استدعاء الاستعلام وتحديد حقل الفرز**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة باسم العمود المراد التقسيم بناءً عليه، ثم الدالة التجميعية المستهدفة لحساب عناصر المجموعة مثل department, COUNT(student_id) .
2. **تحديد مستودع البيانات الأساسي**: الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM بأحرف كبيرة متبوعة مباشرة بالاسم الفعلي للجدول مثل students
3. **استدعاء أداة تجميع الفئات**: الانتقال لسطر جديد وكتابة العبارة المحجوزة GROUP BY بأحرف كبيرة لإعلام المحرك بنمط الفرز التجميعي.

```
SELECT department, COUNT(student_id)
FROM students
GROUP BY department ;
```

4. **تحديد حقل التصنيف الحاكم**: ترك مسافة فارغة واحدة بعد الأداة، ثم إعادة كتابة اسم العمود المحدد لتقسيم المجموعات بناءً عليه مثل department
5. **إغلاق جملة التجميع وعرض التقارير**: إنهاء السكربت بالكامل بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار أمر الفرز وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT (اسم العمود, اسم الدالة(اسم العمود **FROM** اسم الجدول **GROUP BY** اسم العمود ;

```
SELECT department, COUNT(student_id) FROM students GROUP BY department ;
```

لتكون مخرجات التقرير أكثر احترافية وسهلة القراءة للمستخدم النهائي، يُنصح دائماً باستخدام أداة التسمية المستعارة (AS) لتوضيح عمود الإحصاء، هكذا:

```
SELECT department, COUNT(student_id) AS 'Total Students'
FROM students
GROUP BY department ;
```

صياغة شروطاً متقدمة لتصفية المجموعات الإحصائية باستخدام الجملة الشرطية(HAVING)

تُعد هذه المهارة الأداة البرمجية المتقدمة للتحكم في مخرجات التقرير الإحصائي، حيث يتمثل دورها الوظيفي في فرض قيود وشروط صارمة على المجموعات الفرعية الناتجة من عملية التجميع، لاستبعاد الفئات غير المستهدفة. وتكمن الأهمية التعليمية والعملية لتطبيق جملة HAVING في كونها المعامل الشرطي الوحيد القادر على فرز البيانات ومقارنتها بناءً على نتائج الدوال التجميعية، وهو ما تعجز عنه الجملة الشرطية العادية WHERE(مثل عرض الأقسام الأكاديمية التي يزيد عدد طلابها عن 50 طالباً فقط، أو عرض المنتجات التي يتجاوز إجمالي مبيعاتها حداً معيناً)؛ وتعتمد آلية عملها على فحص المجموعات بعد تشكلها تماماً في الذاكرة وتمرير ما يطابق الشرط الحسابي منها.

خطوات كتابة وتطبيق الشرط المتقدم على المجموعات:(HAVING)

1. **استدعاء أمر التجميع الفئوي الكامل**: تبدأ الجملة بصياغة أمر الاستعلام التجميعي الأساسي متضمناً الحقل والدالة وجدول المصدر متبوعاً بعبارة الفرز (مثل كتابة سكربت SELECT department, COUNT(student_id) FROM students GROUP BY department).
2. **استدعاء أداة التصفية المتقدمة للمجموعات**: الانتقال لسطر جديد بعد جملة التجميع وكتابة الكلمة المفتاحية HAVING بأحرف كبيرة.

3. **تحديد الدالة الإحصائية الخاضعة للشرط**: ترك مسافة فارغة واحدة، ثم إعادة كتابة الدالة التجميعية المراد تقييد نتائجها مثل دالة العد COUNT(student_id) .
4. **صياغة القيمة المعيارية المستهدفة**: وضع معامل المقارنة الرياضي متبوعاً بالرقم الحاكم لعرض الفئة أو حجبها مثل وضع شرط الأغلبية 5 < .

```
SELECT department, COUNT(student_id)
FROM students
GROUP BY department
HAVING COUNT(student_id) > 5 ;
```

5. **إغلاق مسار جملة التصفية المركبة**: إنهاء السكربت الإحصائي بالكامل بوضع الفاصلة المنقوطة ; لتأكيد جاهزيته للتنفيذ الفوري والمباشر على السيرفر.

SELECT (اسم العمود)اسم الدالة, اسم العمود**FROM** اسم الجدول **GROUP BY** اسم العمود
HAVING اسم الدالة(اسم العمود) معامل المقارنة القيمة المستهدفة ;

SELECT department, **COUNT**(student_id) **FROM** students **GROUP BY** department **HAVING** COUNT(student_id) > 5 ;

ربط بين عدة جداول برمجياً لاسترجاع بيانات متكاملة ومترابطة في استعلام واحد(JOINS)

تُعد هذه المهارة المفهوم الهندسي الأهم في قواعد البيانات العلائقية، حيث يتمثل دورها الوظيفي في دمج واسترجاع البيانات الموزعة على جدولين أو أكثر وعرضها معاً في جدول نتائج واحد بناءً على حقل مشترك يربط بينهما (مثل المفتاح الرئيسي والأجنبي). وتكمن الأهمية التعليمية والعملية لتطبيق معاملات الربط JOINS في منع تشتيت البيانات وتوفير تقارير شاملة ومترابطة للمستخدم (مثل عرض اسم الطالب من جدول الطلاب بجانب مسمى قسمه المخزن في جدول الأقسام)؛ وتعتمد آلية عملها على قيام المحرك بمسح الجداول المحددة في الاستعلام، ومطابقة الصفوف بناءً على شرط الربط، ثم تجميع الحقول المطلوبة وإخراجها ككتلة بيانات موحدة.

خطوات كتابة أمر الربط بين الجداول:(JOINS)

1. **استدعاء أداة الاستعلام وتحديد الحقول**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة بأسماء الأعمدة المراد عرضها، مع ربط كل عمود باسم جدوله لضمان الدقة مثل SELECT students.student_name, departments.department_name.

```
SELECT students.student_name, departments.department_name
FROM students
INNER JOIN departments
ON students.department_id = departments.department_id ;
```

2. **تحديد الجدول الرئيسي الأول**: كتابة الكلمة المفتاحية FROM بأحرف كبيرة في سطر جديد، يليه اسم الجدول الأساسي الأول مثل students .
3. **استدعاء معامل الربط البرمجي**: ترك مسافة فارغة وكتابة نوع أداة الربط المطلوبة بأحرف كبيرة مثل INNER JOIN.
4. **تحديد الجدول الثاني المستهدف**: ترك مسافة فارغة بعد أداة الربط، ثم كتابة اسم الجدول الثاني المراد سحب البيانات التكميلية من داخله مثل departments.
5. **صياغة شرط المطابقة وإغلاق الأمر**: الانتقال لسطر جديد وكتابة الكلمة المحجوزة ON بأحرف كبيرة لفرض الشرط، يليه مساواة حقل المفتاح الأجنبي بالمفتاح الرئيسي مثل (students.department_id = departments.department_id) ثم إنهاء السكربت بالفاصلة المنقوطة;.

اسم الجدول الأول نوع **FROM** اسم الجدول الأول. اسم العمود, اسم الجدول الثاني. اسم العمود **SELECT** ; اسم الجدول الثاني. حقل الربط =اسم الجدول الأول. حقل الربط **ON** اسم الجدول الثاني **JOIN** الربط

SELECT students.student_name, departments.department_name **FROM** students **INNER JOIN** department;

لتجنب التكرار الطويل لأسماء الجداول وتسهيل قراءة الكود، يمكن استخدام **الأسماء المستعارة للجداول (Table Aliases)** كالتالي:

```
SELECT s.student_name, d.department_name
FROM students s
INNER JOIN departments d
ON s.department_id = d.department_id ;
```

بناء الاستعلامات الفرعية المتداخلة Subqueries لاستخراج بيانات مشروطة بناءً على نتائج استعلام آخر

تُعد هذه المهارة الأسلوب البرمجي المتقدم لتداخل الاستعلامات، حيث يتمثل دورها الوظيفي في بناء جملة استعلام داخلية (Subquery) مغلقة محاطة بأقواس، يتم تنفيذها أولاً لتعود بنتيجة محددة، ثم يمرر محرك القواعد هذه النتيجة تلقائياً لتصبح هي المعيار أو الشرط الحاكم لتنفيذ جملة الاستعلام الخارجية والأساسية. وتكمن الأهمية التعليمية والعملية لتطبيق الاستعلامات الفرعية المتداخلة في تمكين الطالب من حل المسائل البرمجية المعقدة المكونة من خطوتين في خطوة واحدة دون مغادرة السيرفر (مثل استخراج بيانات الطلاب الذين حصلوا على درجات أعلى من متوسط درجات الدفعة بالكامل)؛ وتعتمد آلية عملها على تدرج المعالجة من الداخل إلى الخارج لضمان دقة استخلاص السجلات المشروطة ديناميكياً.

خطوات كتابة وتأسيس الاستعلامات الفرعية المتداخلة:(Subqueries)

1. **صياغة الاستعلام الخارجي الأساسي**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة بالحقول المراد عرضها ثم تحديد الجدول الأساسي وعبارة التصفية الشرطية WHERE مثل SELECT * FROM students WHERE grade_score > .

```
SELECT * FROM students
WHERE grade_score > (SELECT AVG(grade_score) FROM students) ;
```

2. **تهيئة موقع التداخل** :ترك مسافة فارغة واحدة بعد معامل المقارنة الشرطي في الاستعلام الخارجي، ثم فتح قوس هلالي)إيذاناً ببدء بناء السكربت الداخلي.
3. **كتابة الاستعلام الفرعي الداخلي** :صياغة جملة استعلام كاملة ومستقلة تماماً بين الأقواس تبدأ بـ SELECTلاستخراج القيمة المعيارية المستهدفة (مثل SELECT AVG(grade_score) FROM students) .
4. **حصر الاستعلام وإغلاق الأقواس** :وضع قوس هلالي مغلق (فور انتهاء أمر الاستعلام الداخلي مباشرة لتأكيد عزله وتحديد أولوية تنفيذه أولاً من قِبل المحرك.
5. **إنهاء الاستعلام المتداخل بالكامل** :وضع الفاصلة المنقوطة ;في نهاية السطر البرمجي للإعلان عن اكتمال مسار الاستعلام المركب وجاهزيته للتنفيذ الفوري على السيرفر.

SELECT اسم العمود **FROM**اسم الجدول **WHERE** اسم العمود معامل المقارنة
(**SELECT** (اسم العمود)دالة تجميعية **FROM** (اسم الجدول) ;

```
SELECT * FROM students WHERE grade_score > (SELECT AVG(grade_score)
FROM students) ;
```

الربط التبادلي لاسترجاع الجداء الديكارتي لجميع السجلات بين جدولين باستخدام(CROSS JOIN)

تُعد هذه المهارة الأداة البرمجية المخصصة لإنتاج التباديل والتوافيق الاحتمالية الكاملة بين البيانات، حيث يتمثل دورها الوظيفي في دمج كل سجل من سجلات الجدول الأول (الأيسر) مع كافة سجلات الجدول الثاني (الأيمن) دون التقيد بأي حقل مشترك أو شرط ربط. وتكمن الأهمية التعليمية والعملية لتطبيق الربط التبادلي CROSS JOIN في تدريب الطالب على توليد المصفوفات التمريرية وتوليف العينات (مثل ربط جدول يحتوي على تخصصات الطلاب مع جدول آخر يحتوي على مستويات الصفوف الدراسية لإنتاج كافة الاحتمالات الممكنة للشعب)؛ وتعتمد آلية عملها على مضاعفة حجم البيانات المسترجعة، فإذا كان الجدول الأول يحتوي على 3 سجلات والثاني يحتوي على 4 سجلات، فإن الناتج النهائي سيكون حاصل ضربهما وهو 12 سجلاً بالتمام والكمال.

خطوات كتابة أمر الربط التبادلي:(CROSS JOIN)

1. **استدعاء أداة الاستعلام وتحديد عناصر العرض**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECTمتبوعة بالحقول المراد عرضها من كلا الجدولين مثل * SELECT لعرض كافة الأعمدة المتولدة.

```
SELECT * FROM departments
CROSS JOIN courses ;
```

2. **تحديد الجدول الأيسر الأول**: الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM بأحرف كبيرة متبوعة مباشرة باسم الجدول الأساسي الأول مثل departments .
3. **استدعاء معامل الربط التبادلي الحاكم**: ترك مسافة فارغة واحدة، ثم كتابة العبارة المحجوزة CROSS JOINبأحرف كبيرة لتوجيه السيرفر نحو دمج السجلات تبادلياً.
4. **تحديد الجدول الأيمن الثاني**: ترك مسافة فارغة بعد أداة الربط، ثم كتابة المسمى الصريح للجدول الثاني المراد ضرب سجلاته مثل courses.
5. **إغلاق مسار الجملة الاستعلامية التبادلية**: إنهاء السكربت بالكامل بوضع الفاصلة المنقوطة ;مباشرة بعد اسم الجدول الثاني (دون كتابة كلمة ONأو صياغة شروط) للإشارة إلى جاهزية السكربت للتنفيذ الفوري على السيرفر.

SELECT اسم العمود **FROM** اسم الجدول الأول **CROSS JOIN** اسم الجدول الثاني ;

```
SELECT departments.department_name, courses.course_name FROM departments
CROSS JOIN courses ;
```

استخراج السجلات المشتركة والمطابقة تماماً بين جدولين باستخدام صيغة الربط الداخلي(INNER JOIN)

تُعد هذه المهارة الأداة البرمجية الأكثر استخداماً لتصفية البيانات المترابطة بين جدولين، حيث يتمثل دورها الوظيفي في استخراج وعرض الصفوف التي تحتوي على قيم متطابقة تماماً في الحقل المشترك بين الجدولين (مثل المفتاح الرئيسي والأجنبي) وإهمال أي سجلات أخرى لا تقابلها بيانات في الطرف الثاني. وتكمن الأهمية التعليمية والعملية لتطبيق الربط الداخلي في دقة المخرجات وتقييدها بالبيانات الفعلية المكتملة فقط (مثل عرض أسماء الطلاب الذين لديهم أقسام مسجلة بالفعل، واستبعاد أي طالب لم يتم تحديد قسمه

بعد)؛ وتعتمد آلية عملها على قيام المحرك بتقاطع السجلات بين المصدرين وعرض مساحة التداخل المشتركة بينهما فقط.

خطوات كتابة أمر الربط الداخلي:(INNER JOIN)

1. **استدعاء أداة الاستعلام وتوجيه الحقول**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة بالحقول المنتقاة من كلا الجدولين مع الفصل بينها بوضع اسم الجدول متبوعاً بنقطة قبل اسم العمود مثل SELECT students.student_name, sections.section_name .

```
SELECT students.student_name, sections.section_name
FROM students
INNER JOIN sections
ON students.section_id = sections.section_id ;
```

2. **تحديد الجدول الأيسر الأول**: الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM بأحرف كبيرة يليه اسم الجدول الأساسي الأول مثل students.
3. **استدعاء معامل الفرز الداخلي**: ترك مسافة فارغة واحدة، ثم كتابة العبارة المحجوزة INNER JOIN بأحرف كبيرة لتوجيه السيرفر نحو استخراج المتطابقات فقط.
4. **تحديد الجدول الأيمن الثاني**: ترك مسافة فارغة بعد أداة الربط، ثم كتابة اسم الجدول التكميلي الثاني مثل sections.
5. **فرض شرط التطابقة وإغلاق الأمر**: الانتقال لسطر جديد وكتابة الكلمة المحجوزة ON بأحرف كبيرة يليه مساواة حقول الربط المشتركة بين الجدولين مثل students.section_id = sections.section_id ثم إنهاء السكربت بالفاصلة المنقوطة;.

SELECT اسم العمود. اسم الجدول الثاني. اسم العمود، اسم الجدول الأول. **FROM** اسم الجدول الأول **INNER JOIN** اسم الجدول الثاني **ON** حقل الربط =اسم الجدول الأول. حقل الربط اسم الجدول الثاني. ;

SELECT students. student_name, sections.section_name **FROM** students **INNER JOIN** sections **ON** student;

لتسهيل كتابة الكود وجعله أكثر احترافية وقابلية للقراءة، يمكنك استخدام **الأسماء المستعارة للجداول (Table Aliases)** لاختصار أسماء الجداول الطويلة، هكذا:

```
SELECT s.student_name, sec.section_name
FROM students s
INNER JOIN sections sec
ON s.section_id = sec.section_id ;
```

الربط الخارجي الأيسر لاسترجاع سجلات الجدول بالكامل حتى لو لم تقابلها بيانات بالطرف الآخر باستخدام (LEFT JOIN)

تُعد هذه المهارة الأداة البرمجية المخصصة للاحتفاظ بكافة بيانات الطرف الأول، حيث يتمثل دورها الوظيفي في استرجاع جميع السجلات المخزنة داخل الجدول الأيسر الجدول المكتوب أولاً بعد جملة FROM بالكامل وبصورة إلزامية، مع سحب البيانات المطابقة لها فقط من الجدول الأيمن، ووضع قيمة فارغة NULL في حقول الجدول الأيمن التي لا تجد سجلات تقابلها. وتكمن الأهمية التعليمية والعملية لتطبيق LEFT JOIN في ضمان عرض الهيكل الأساسي للنظام دون نقصان (مثل عرض قائمة بجميع الأقسام الأكاديمية بالكامل حتى لو كان هناك قسم جديد لم يسجل به أي طالب بعد)؛ وتعتمد آلية عملها على جعل الجدول الأيسر هو المحور الثابت للاستعلام ودمج البيانات التكميلية معه بمرونة.

خطوات كتابة أمر الربط الخارجي الأيسر:(LEFT JOIN)

1. **استدعاء أداة الاستعلام وتنسيق الحقول**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة بالحقول المراد استخراجها مع ربط كل حقل باسم جدوله (مثل SELECT departments.department_name, students.student_name).
2. **تحديد الجدول الأيسر الأساسي**: الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM متبوعة باسم الجدول الأول الذي يقع جهة اليسار ويمثل الأساس مثل departments.

```
SELECT departments.department_name, students.student_name
FROM departments
LEFT JOIN students
ON departments.department_id = students.department_id ;
```

3. **استدعاء معامل الربط الأيسر**: ترك مسافة فارغة واحدة، ثم كتابة العبارة المحجوزة LEFT JOIN بأحرف كبيرة للاحتفاظ بكامل الجدول الأيسر.
4. **تحديد الجدول الأيمن التكميلي**: ترك مسافة فارغة بعد أداة الربط، ثم كتابة اسم الجدول الثاني الذي يقع جهة اليمين مثل students.
5. **فرض شرط الربط وإغلاق الأمر**: الانتقال لسطر جديد وكتابة الكلمة المحجوزة ON متبوعة بمساواة الحقول المشتركة مثل departments.department_id = students.department_id ثم إنهاء السكربت بالفاصلة المنقوطة; .

SELECT اسم العمود. اسم الجدول الثاني, اسم العمود. اسم الجدول الأول **FROM** اسم الجدول الأول **LEFT JOIN** اسم الجدول الثاني **ON** حقل الربط. اسم الجدول الأول= حقل الربط. اسم الجدول الثاني ;

```
SELECT departments.department_name, students.student_name FROM departments
LEFT JOIN students ON departments.department_id = students.department_id ;
```

باستخدام **الأسماء المستعارة للجداول(Table Aliases)** ، يمكنك اختصار الكود وجعله أكثر نظافة وقابلية للقراءة السريعة، هكذا:

```
SELECT d.department_name, s.student_name
FROM departments d
LEFT JOIN students s
ON d.department_id = s.department_id ;
```

الربط الخارجي الأيمن لاسترجاع سجلات الجدول بالكامل حتى لو لم تقابلها بيانات بالطرف الآخر باستخدام (RIGHT JOIN)

تُعد هذه المهارة الأداة البرمجية العكسية المخصصة للاحتفاظ بكافة بيانات الطرف الثاني، حيث يتمثل دورها الوظيفي في استرجاع جميع السجلات المخزنة داخل الجدول الأيمن الجدول المكتوب ثانياً بعد جملة RIGHT JOIN بالكامل وبصورة إلزامية، مع سحب البيانات المطابقة لها فقط من الجدول الأيسر، ووضع قيمة فارغة NULL في حقول الجدول الأيسر التي لا تجد سجلات تقابلها. وتكمن الأهمية التعليمية والعملية لتطبيق RIGHT JOIN في فرز وتحليل البيانات من منظور الجدول التكميلي مثل عرض قائمة بجميع

الطلاب بالكامل حتى لو كان هناك طالب مستجد لم يتم توزيعه أو ربطه بقسم أكاديمي بعد)؛ وتعتمد آلية عملها على جعل الجدول الأيمن هو المحور الثابت للاستعلام.

خطوات كتابة أمر الربط الخارجي الأيمن:(RIGHT JOIN)

1. **استدعاء أداة الاستعلام وتنسيق الحقول**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة بالحقول المراد استخراجها مع ربط كل حقل باسم جدوله مثل SELECT departments.department_name, students.student_name .
2. **تحديد الجدول الأيسر**: الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM متبوعة باسم الجدول الأول مثل departments .
3. **استدعاء معامل الربط الأيمن الحاكم**: ترك مسافة فارغة واحدة، ثم كتابة العبارة المحجوزة RIGHT JOIN بأحرف كبيرة للاحتفاظ بكامل الجدول الأيمن.

```
SELECT d.department_name, s.student_name
FROM departments d
RIGHT JOIN students s
ON d.department_id = s.department_id ;
```

4. **تحديد الجدول الأيمن الأساسي**: ترك مسافة فارغة بعد أداة الربط، ثم كتابة اسم الجدول الثاني الذي يقع جهة اليمين ويمثل المحور هنا مثل students .
5. **فرض شرط الربط وإغلاق الأمر**: الانتقال لسطر جديد وكتابة الكلمة المحجوزة ONمتبوعة بمساواة الحقول المشتركة مثل departments.department_id = students.department_id ثم إنهاء السكربت بالفاصلة المنقوطة; .

SELECT اسم العمود. اسم الجدول الثاني، اسم العمود. اسم الجدول الأول **FROM** اسم الجدول الأول **RIGHT JOIN** اسم الجدول الثاني **ON** حقل الربط. اسم الجدول الأول= حقل الربط. اسم الجدول الثاني ;

SELECT departments. department_name, students. student_name FROM departments **RIGHT JOIN** students **ON** departments. department_id = students.department_id ;

الربط الذاتي SELF JOINلمقارنة وسحب سجلات من نفس الجدول باعتباره جدولين منفصلين

تُعد هذه المهارة الأسلوب البرمجي الذكي لمعالجة البيانات ذات العلاقات الداخلية، حيث يتمثل دورها الوظيفي في ربط الجدول بنسخة ثانية من نفسه برمجياً لاستخراج سجلات ومقارنتها ببعضها البعض. وتكمن الأهمية التعليمية والعملية لتطبيق الربط الذاتي SELF JOIN في تمكين الطالب من حل المشكلات الهيكلية المعقدة داخل الكيان الواحد (مثل جدول الموظفين الذي يحتوي على عمود لمعرّف المدير وهو في الأصل موظف داخل نفس الجدول، أو مقارنة درجات الطلاب ببعضهم في نفس الحقل)؛ وتعتمد آلية عملها على خداع محرك البحث عبر تكرار كتابة اسم الجدول مرتين في سطر الربط مع منحهما اسمين مستعارين مختلفين كأنهما جدولان مستقلان تماماً.

خطوات كتابة أمر الربط الذاتي:(SELF JOIN)

1. **استدعاء أداة الاستعلام وتحديد الأسماء المستعارة للحقول**: تبدأ الجملة البرمجية بكتابة الكلمة المحجوزة SELECT متبوعة بالحقول المراد عرضها، مع ربط كل حقل بالحرف الرمزي المستعار لنسخته مثل SELECT a.student_name, b.student_name.
2. **تحديد الجدول الأصلي وتعيين الرمز الأول**: الانتقال لسطر جديد وكتابة الكلمة المفتاحية FROM متبوعة باسم الجدول الفعلي، ثم ترك مسافة وكتابة حرف رمزي يمثل النسخة الأولى مثل FROM students a .
3. **استدعاء معامل الربط وتعيين الرمز الثاني**: ترك مسافة فارغة وكتابة أداة الربط مثل INNER JOIN متبوعة بإعادة كتابة نفس اسم الجدول الفعلي مرة أخرى، ثم ترك مسافة وكتابة حرف رمزي مختلف يمثل النسخة الثانية مثل INNER JOIN students b.
4. **صياغة شرط المقارنة الداخلي**: الانتقال لسطر جديد وكتابة الكلمة المحجوزة ON متبوعة بمعادلة الربط والمقارنة بين النسختين مثل مساواة رقم كود المدير برقم كود الموظف a.manager_id = b.employee_id .

```
SELECT a.employee_name AS 'Employee', b.employee_name AS 'Manager'
FROM employees a
INNER JOIN employees b
ON a.manager_id = b.employee_id ;
```

5. **إغلاق جملة الاستعلام وعرض النتائج**: إنهاء السكربت بالكامل بوضع الفاصلة المنقوطة ; للإشارة إلى نهاية مسار الأمر وجاهزيته للتنفيذ الفوري على السيرفر.

في أوامر الـ Self Join، نحن لا نقوم بإنشاء جداول حقيقية جديدة، بل نستخدم **الأسماء المستعارة a و b** لإنشاء نسختين افتراضيتين من نفس الجدول في الذاكرة المؤقتة، مما يسمح للسيرفر بمقارنة صفوف الجدول ببعضها البعض بسهولة.

SELECT اسم العمود. اسم العمود، الرمز الثاني. الرمز الأول **FROM** اسم الجدول الرمز الأول نوع الربط **JOIN** اسم الجدول الرمز الثاني **ON** حقل الربط = الرمز الأول. حقل الربط الرمز الثاني ;

```
SELECT a. employee_name AS Employee, b. employee_name AS Manager FROM
employees a INNER JOIN employees b ON a. manager_id = b. employee_id ;
```

إنشاء الجداول الافتراضية والمرجعية الثابتة لحفظ وتأمين الاستعلامات المعقدة والمتكررة باستخدام الأمر (CREATE VIEW)

تُعد هذه المهارة الأداة البرمجية المتقدمة لحماية وتسهيل الوصول إلى البيانات، حيث يتمثل دورها الوظيفي في حفظ استعلام معقد ومطول (مثل استعلام يحتوي على روابط متعددة ودوال تجميعية) وتخزينه داخل قاعدة البيانات كجدول افتراضي وهمي (View) يمكن استدعاؤه بكلمة واحدة في أي وقت. وتكمن الأهمية التعليمية والعملية لتطبيق الجداول الافتراضية في شقين: الأول هو تبسيط العمليات المتكررة للمطورين، والثاني هو تأمين البيانات الحساسة؛ حيث تتيح للطالب إظهار حقول محددة فقط للمستخدمين دون كشف الأعمدة السرية أو البنية الفيزيائية الحقيقية للجداول الأساسية؛ وتعتمد آلية عملها على تحديث بيانات الجدول الافتراضي ديناميكياً وتلقائياً فور حدوث أي تعديل في الجداول الأصلية.

خطوات كتابة أمر إنشاء الجدول الافتراضي:(CREATE VIEW)

1. **استدعاء أداة التأسيس الافتراضي**: تبدأ الجملة البرمجية بكتابة العبارة المحجوزة CREATE VIEW بأحرف كبيرة للإعلان عن بناء كائن مرجعي جديد داخل قاعدة البيانات.
2. **تسمية الجدول الافتراضي**: ترك مسافة فارغة واحدة، ثم كتابة اسم مميز ومعبر للـ View الجديد لتسهيل استدعائه لاحقاً مثل v_top_students

3. **تضمين أداة الربط البرمجي**: ترك مسافة فارغة وكتابة الكلمة المفتاحية المحجوزة AS بأحرف كبيرة لربط المسمى بالاستعلام التنفيذي.
4. **صياغة الاستعلام المستهدف بالدوران**: الانتقال لسطر جديد وكتابة جملة الاستعلام القياسية بالكامل والمطلوب حفظها مثل صياغة أمر SELECT * FROM students WHERE grade_score >= 90 .
5. **إغلاق جملة التأسيس الهيكلية**: إنهاء السكربت بالكامل بوضع الفاصلة المنقوطة ; في نهاية جملة الاستعلام لإعلام السيرفر باكتمل بناء الجدول الافتراضي وجاهزيته للاستخدام.

```
CREATE VIEW v_top_students AS
SELECT * FROM students
WHERE grade_score >= 90 ;
```

CREATE VIEW اسم الجدول الافتراضي **AS SELECT** اسم العمود **FROM** اسم الجدول **WHERE** الشرط الحاكم ;

```
CREATE VIEW v_top_students AS SELECT student_name, grade_score FROM students WHERE grade_score >= 90 ;
```

بعد إنشاء هذا الجدول الافتراضي بنجاح، لن تحتاج إلى كتابة الاستعلام الطويل مرة أخرى، بل يمكنك ببساطة استدعاء البيانات منه وكأنه جدول عادي جداً، هكذا:

```
SELECT * FROM v_top_students ;
```