02 — الجزء الثاني

دليل SQL العملي للمحللين

من الجداول إلى القرار

استعلام — تحليل — قرار



عبدالرحمن الأكلبي
aalaklabi.com

المحتويات

- قبل أن تبدأ 3
- الفصل الخامس — اربط الجداول دون تضخيم الأرقام 5
- الفصل السادس — حضّر بيانات قابلة للتحليل 16
- الفصل السابع — تحقق قبل أن ترسل الرقم 22
- الفصل الثامن — مشروع مصغّر: أداء الطلبات والمبيعات 26
- المرجع السريع 32
- حلول التمارين 33
- المسرد والمصادر 37
- عن الكاتب والخاتمة 38

قبل أن تبدأ

يكمل SQL 02 الاختيار والتصفية والتجميع بالربط و`CASE` وCTE وبناء المؤشرات. يلزم فهم `SELECT` و`WHERE` و `GROUP BY` ووحدة الصف.

الفصول 5–8 تكمل المسار. البيانات نفسها؛ لا تعِد تحميلها إن كانت جاهزة. حلول E05–E08 في نهاية الجزء.

بياناتنا: «متاجر واحة»

بيانات «متاجر واحة» **افتراضية** لثمانية فروع، خلال 2024–2025، وتتضمن إلغاءات وقيمًا ناقصة للتدريب. المبالغ بالريال دون الضريبة. أوقات الطلبات محلية للرياض (UTC+3) دون منطقة زمنية في الحقل. `list_price` سعر الرف الحالي في بيانات 2025، وليس سعر البيع التاريخي.

الجداول المستخدمة ووحدة الصف:

الجدول	ماذا يحتوي؟	ماذا يمثل الصف الواحد؟
`products`	المنتجات وأسعار الرف والفئات	منتج واحد
`categories`	فئات المنتجات	فئة واحدة
`orders`	الطلبات والتاريخ والفرع والحالة	طلب واحد
`order_items`	المنتجات والكميات والأسعار داخل كل طلب	سطرًا واحدًا في طلب
`branches`	أسماء الفروع ومدنها	فرع واحد
`customers`	تسجيلات برنامج الولاء	تسجيل عميل، وقد يتكرر الشخص بأكثر من تسجيل

إعداد التطبيق

04

فك حزمة `sql-guide-companion.zip` وافتح مجلد `dataset`. تفاصيل الإعداد والملفات في `SETUP.md` بالحزمة المصاحبة؛ تعريف الحقول في `DATA_DICTIONARY.md`.

> شغّل **PostgreSQL 16** وجهّز `createdb` و`psql`. المستخدم في الأوامر `postgres`؛ استخدم بيانات اتصالك.

> من طرفية مجلد **`dataset`**، أنشئ قاعدة تدريب جديدة وحمّل البيانات:

```
createdb -U postgres waha_practical_v02
psql -U postgres -d waha_practical_v02 -v ON_ERROR_STOP=1 -f load.psql
```

تنبيه: `schema.sql` يحذف الجداول السابقة؛ استخدم قاعدة تدريب جديدة. إذا كان الاسم مستخدمًا، غيّره في الأمرين. شغّل `load.psql` من `dataset` عبر **psql**، لا داخل محرر pgAdmin.

> تحقق من **34 منتجًا، و31,474 طلبًا، و95,281 سطر طلب**، وصفر مخالفات في الفحوص الخمسة. عند الفشل، راجع أول خطأ.

> افتح جلسة الاستعلام، أو اتصل بالقاعدة في pgAdmin:

```
psql -U postgres -d waha_practical_v02
```

التثبيت وخطوات التشغيل المفصلة في `SETUP.md`؛ المراجع الرسمية: psql وcreatedb.

05

اربط الجداول دون تضخيم الأرقام

سؤال الفصل:

كيف تعرض مبيعات كل فرع مع اسمه، دون أن تخلط عدد الطلبات بعدد أسطرها؟

في هذا الفصل

5.1 لماذا نستخدم JOIN؟

5.2 مثال غذائي: المنتج وفئته

5.3 العلاقة التي تضاعف الصفوف

5.4 احسب المبيعات بتعريف واضح

5.5 ماذا لو لم يكن للفرع طلبات؟

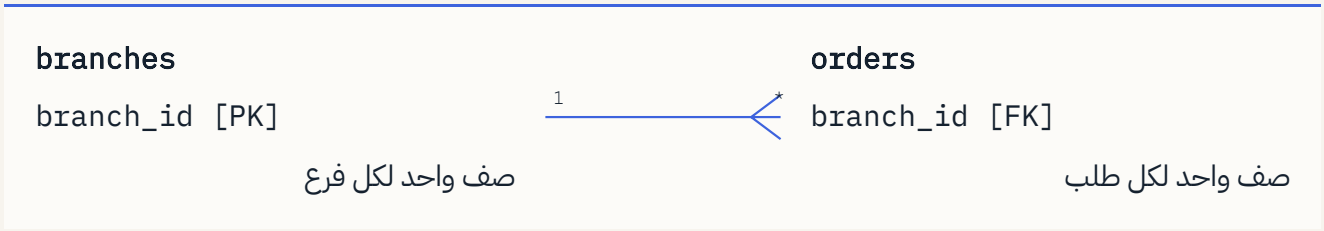
اربط الجداول دون تضخيم الأرقام

نربط الطلبات بالفروع لعرض أسمائها، وبأسطرها لحساب المبلغ. افهم علاقة الجداول قبل الربط.

5.1 لماذا نستخدم JOIN؟

كنا نرى `branch_id = 1` في النتائج. الآن نريد رمز الفرع ومدينته إلى جانب عدد طلباته.

المفتاح الأساسي (Primary Key) يعرّف صفًا دون تكرار؛ `branches.branch_id` يعرّف الفرع. **المفتاح الأجنبي (Foreign Key)** يشير إلى هذا المفتاح من جدول آخر؛ `orders.branch_id` يشير إلى فرع الطلب. **علاقة واحد إلى عدة** تعني أن الفرع الواحد قد يرتبط بعدة طلبات.



`1` و`*` تعنيان جهة واحدة وجهة متعددة، لا قيمًا مخزنة.

`JOIN`، أو `INNER JOIN`، يختار الصفوف المتطابقة بحسب شرط `ON`. كل طلب في المصدر يحمل معرّف فرع صالحًا، وكل معرّف في جدول الفروع فريد؛ لذلك يظل الطلب صفًا واحدًا بعد هذا الربط:

5.1 لماذا نستخدم `JOIN`؟

```sql
SELECT
  b.branch_code,
  b.city,
  COUNT(*) AS completed_orders
FROM orders o
JOIN branches b ON b.branch_id = o.branch_id
WHERE o.status = 'completed'
  AND o.order_ts >= '2025-01-01'
  AND o.order_ts < '2026-01-01'
GROUP BY b.branch_code, b.city
ORDER BY completed_orders DESC, b.branch_code;
```

أعلى ثلاثة صفوف:

branch_code	city	completed_orders
RYD-01	الرياض	2,778
JED-01	جدة	2,661
RYD-02	الرياض	2,158

يعرض الجدول جزءًا من النتيجة، لا جميع الصفوف.

`o` اختصار `orders` و`b` اختصار `branches`. يطابق `ON` معرّف الفرع؛ كتابة `o.branch_id` تحدد الجدول عند تكرار اسم الحقل.

5.2 مثال غذائي: المنتج وفئته

بعض المعلومات تظهر في جدول المنتجات، وبعضها الآخر في جدول الفئات. لنقرأ جميع منتجات التمور والقهوة مع اسم الفئة:

```sql
SELECT
  p.product_name,
  c.category_name,
  p.list_price
FROM products p
JOIN categories c ON c.category_id = p.category_id
WHERE p.category_id = 6
ORDER BY p.list_price DESC, p.product_id;
```

جزء من النتيجة:

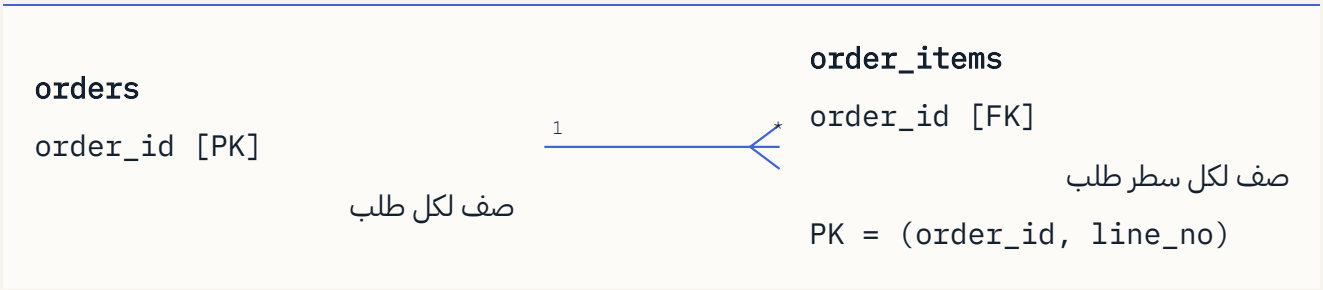
product_name	category_name	list_price
تمر عجوة 1 كجم	تمور وقهوة	60.00
هيل حب 250 جم	تمور وقهوة	44.00
قهوة عربية 500 جم	تمور وقهوة	38.00
قهوة مختصة مطحونة 250 جم	تمور وقهوة	36.00

يعرض الجدول جزءًا من النتيجة، لا جميع الصفوف.

كل منتج يرتبط بفئة واحدة، ومعرّف الفئة فريد؛ لذلك يبقى المنتج صفًا واحدًا بعد الربط.

5.3 العلاقة التي تضاعف الصفوف

أما العلاقة بين `orders` و`order_items` فهي مختلفة:



في `order_items`، المفتاح الأساسي مركب من معرّف الطلب ورقم السطر معًا. يتكرر معرّف الطلب بصورة طبيعية، لكن الزوج لا يتكرر. لنرَ أسطر **طلب فعلي في حزمة التدريب**:

```sql
SELECT
  o.order_number,
  oi.line_no,
  p.product_name
FROM orders o
JOIN order_items oi ON oi.order_id = o.order_id
JOIN products p ON p.product_id = oi.product_id
WHERE o.order_id = 14516
ORDER BY oi.line_no;
```

order_number	line_no	product_name
W0014516	1	سكر أبيض 2 كجم
W0014516	2	مشروب غازي 6 علب
W0014516	3	لبن 1 لتر

ظهور رقم الطلب في كل سطر صحيح؛ النتيجة تعرض أسطره، ولا تعني وجود عدة طلبات. `COUNT(*)` على هذا الناتج يعد الأسطر، و`COUNT(DISTINCT o.order_id)` يعد الطلب مرة واحدة.

5.3 العلاقة التي تضاعف الصفوف

لنختبر الفرق على طلبات 2025 المكتملة:

```sql
SELECT
  COUNT(*) AS item_rows,
  COUNT(DISTINCT o.order_id) AS completed_orders
FROM orders o
JOIN order_items oi ON oi.order_id = o.order_id
WHERE o.status = 'completed'
  AND o.order_ts >= '2025-01-01'
  AND o.order_ts < '2026-01-01';
```

item_rows	completed_orders
50,079	16,585

50,079 صفًا بعد الربط، لكنها تخص **16,585** طلبًا مختلفًا. إذا كنت تريد عدد الطلبات، استخدم `COUNT(DISTINCT o.order_id)` في هذه الحالة. وإذا أردت عدد أسطر المنتجات، فـ `COUNT(*)` هو الصحيح. لا تستخدم `DISTINCT` عشوائيًا لإخفاء تضاعف ناتج؛ تأكد أولًا من وحدة القياس.

5.4 احسب المبيعات بتعريف واضح

بما أن السعر والكمية والخصم في أسطر الطلبات، فإن المبلغ الأساسي لكل سطر هو:

`quantity × unit_price − discount_amount`

- `quantity`: عدد الوحدات في السطر.
- `unit_price`: سعر الوحدة الفعلي وقت البيع، قبل الخصم؛ نستخدمه للمبيعات التاريخية بدل سعر الرف الحالي `products.list_price`.
- `discount_amount`: خصم على **السطر كله**، نطرحه مرة واحدة بعد ضرب الكمية في السعر. ليس خصمًا لكل وحدة.

5.4 احسب المبيعات بتعريف واضح

- `SUM`: تجمع مبالغ الأسطر المختارة. `ROUND(..., 2)` تقرّب الناتج النهائي إلى منزلتين للعرض؛ لا نقرّب كل سطر قبل الجمع.

ثم نجمع السطور **للطلبات المكتملة فقط** ونربط رمز الفرع:

```sql
SELECT
  b.branch_code,
  ROUND(
    SUM(oi.quantity * oi.unit_price - oi.discount_amount),
    2
  ) AS sales_before_returns
FROM orders o
JOIN order_items oi ON oi.order_id = o.order_id
JOIN branches b ON b.branch_id = o.branch_id
WHERE o.status = 'completed'
  AND o.order_ts >= '2025-01-01'
  AND o.order_ts < '2026-01-01'
GROUP BY b.branch_code
ORDER BY sales_before_returns DESC, b.branch_code;
```

أعلى ثلاثة فروع بحسب مجموع مبالغ السطور:

branch_code	sales_before_returns
RYD-01	284,948.58
JED-01	270,755.79
RYD-02	220,883.79

يعرض الجدول جزءًا من النتيجة، لا جميع الصفوف.

4 . 5 احسب المبيعات بتعريف واضح

المؤشر **المبيعات بعد خصومات السطور وقبل المرتجعات** (`K3`): `K1` مجموع الكمية × السعر، و`K2` مجموع خصومات السطور، و`K3 = K1 − K2`. الصافي بعد المرتجعات (`K6`) يطرح مبالغ الإرجاع (`K4`). تُنسب المبيعات لتاريخ الطلب والمرتجعات لتاريخ الإرجاع وفرع الفاتورة الأصلية؛ قد يرجع بيع من فترة أسبق. كل المبالغ دون الضريبة.

ربط المرتجعات مباشرة قد يكرر مبلغ السطر؛ راجع علاقتها بالسطر قبل إضافتها للحساب.

متوسط الطلب قبل استخدامه في المشروع

المتوسط الحسابي لمجموعة قيم هو مجموعها مقسومًا على عددها. متوسط مبلغ الطلب المكتمل هنا هو **مجموع مبالغ الطلبات بعد خصومات السطور وقبل المرتجعات ÷ عدد الطلبات المكتملة**، وهو `K7` في قاموس التدريب.

```sql
SELECT
  ROUND(
    SUM(oi.quantity * oi.unit_price - oi.discount_amount)
      / COUNT(DISTINCT o.order_id),
    2
  ) AS avg_before_returns_per_order
FROM orders o
JOIN order_items oi ON oi.order_id = o.order_id
WHERE o.status = 'completed'
  AND o.order_ts >= '2025-01-01'
  AND o.order_ts < '2026-01-01';
```

avg_before_returns_per_order
102.65

الناتج بالريال لكل طلب. دالة `AVG` تحسب متوسط قيم الصفوف التي تمررها؛ لو طبقتها على مبالغ **الأسطر** لحسبت متوسط السطر، لا متوسط الطلب. لذلك نستخدم القسمة على عدد الطلبات المتميزة. في نطاق المثال توجد طلبات، فلا يكون المقام صفرًا؛ إذا عممت الحساب على فترة فارغة، عالج المقام صفرًا قبل القسمة.

5.5 ماذا لو لم يكن للفرع طلبات؟

`JOIN` العادية تعيد الصفوف التي تجد لها مطابقة. أما `LEFT JOIN` فتُبقي صفوف الجدول الواقع يسارها حتى إن لم تجد مطابقة يمينها. سنستخدم **يناير وفبراير 2025**: فرع المدينة `MED-01` لم يفتتح حتى 1 مارس، ولذلك لا توجد له طلبات في هذه الفترة.

```sql
SELECT
  b.branch_code,
  COUNT(o.order_id) AS completed_orders
FROM branches b
LEFT JOIN orders o ON o.branch_id = b.branch_id
  AND o.status = 'completed'
  AND o.order_ts >= '2025-01-01'
  AND o.order_ts < '2025-03-01'
GROUP BY b.branch_code
ORDER BY b.branch_code;
```

branch_code	completed_orders
AHB-01	257
DMM-01	318
JED-01	397
JED-02	327
MED-01	0
MKK-01	328
RYD-01	416
RYD-02	322

5.5 ماذا لو لم يكن للفرع طلبات؟

بدأنا من `branches` كي تظهر كل الفروع، ووضعنا شروط الطلبات داخل `ON` ليبحث الربط عن طلبات الفترة فقط دون إسقاط الفرع بلا مطابقة. قيم الطلب في صف المدينة تكون `NULL`؛ لذا يعد `COUNT(o.order_id)` صفر طلبات. أما `COUNT(*)` فيعد الصف المحفوظ نفسه فيعطي 1.

قارن بالربط الداخلي في الفترة نفسها:

```sql
SELECT
  b.branch_code,
  COUNT(o.order_id) AS completed_orders
FROM branches b
JOIN orders o ON o.branch_id = b.branch_id
  AND o.status = 'completed'
  AND o.order_ts >= '2025-01-01'
  AND o.order_ts < '2025-03-01'
GROUP BY b.branch_code
ORDER BY b.branch_code;
```

branch_code	completed_orders
AHB-01	257
DMM-01	318
JED-01	397
JED-02	327
MKK-01	328
RYD-01	416
RYD-02	322

5.5 ماذا لو لم يكن للفرع طلبات؟

علامة تحقق: `LEFT JOIN` يعرض ثمانية فروع، منها `MED-01` بصفر؛ `JOIN` يعرض سبعة ويغيب عنه هذا الفرع. أعداد الفروع السبعة متطابقة. الصفر يصف غياب طلبات الفترة، ولا يثبت ضعف أداء فرع لم يكن مفتوحًا. وإذا نقلت شروط الطلبات إلى `WHERE` في مثال `LEFT JOIN` فسوف تستبعد الصف بلا مطابقة؛ احتفظ بها في `ON`.

قاعدة: قبل الربط اسأل: هل هذه العلاقة واحد إلى واحد، أم واحد إلى عدة؟ وما الذي سيمثله الصف الواحد بعد الربط؟

> **تدريب E05 — جرّب بنفسك:** اعرض `branch_name` بدل `branch_code` في ملخص 5.1 مع المدينة وعدد الطلبات. اجمع بحسب `branch_id` والاسم والمدينة لضمان صف لكل فرع، وقارن العدد بنتيجة 4.3. ثم تخيّل ملف فروع مستوردًا خارج القاعدة التدريبية، لا يفرض مفتاحًا فريدًا، وفيه صفان بالمعرّف نفسه. كيف يؤثر ربطه بالطلبات؟ المفتاح الأساسي في `branches` الأصلي يمنع إدخال هذا التكرار؛ السيناريو يخص الملف المستورد فقط.

> **قبل أن تنتقل:** فسّر مفاتيح الربط، والفرق بين `JOIN` و`LEFT JOIN`، ووحدة الناتج وتعريف المبلغ.

06

حضّر بيانات قابلة للتحليل

سؤال الفصل:
كيف تحول أعمدة المصدر إلى أعمدة مفهومة لقارئ التقرير دون تغيير الحقائق؟

في هذا الفصل

6.1 صنّف بقواعد واضحة: CASE
6.2 عالج القيمة الناقصة دون إخفائها
6.3 تقسيم سعري: مثال لا حكم
6.4 سلّم جدولًا يستطيع غيرك قراءته
6.5 سمِّ خطوة الحساب باستخدام CTE

حضّر بيانات قابلة للتحليل

جهّز أسماء وقيمًا مفهومة للتقرير، مع الحفاظ على معنى بيانات المصدر.

6.1 صنّف بقواعد واضحة: CASE

لنفترض أن التقرير العربي يحتاج اسمًا مفهومًا لقناة البيع:

```sql
SELECT
  order_number,
  CASE
    WHEN channel = 'online' THEN 'إلكتروني'
    ELSE 'داخل المتجر'
  END AS sales_channel
FROM orders
WHERE status = 'completed'
  AND order_ts >= '2025-01-01'
  AND order_ts < '2026-01-01'
ORDER BY order_id
LIMIT 5;
```

order_number	sales_channel
W0014515	داخل المتجر
W0014516	داخل المتجر
W0014517	داخل المتجر
W0014518	إلكتروني
W0014519	داخل المتجر

6.1 صنّف بقواعد واضحة: `CASE`

`CASE` تختار القيمة حسب الشرط. لدينا قناتان موثقتان، لذا تمثل `ELSE` «داخل المتجر». إذا أضيفت قناة ثالثة، عدّل القاعدة أو صنفها «أخرى» للمراجعة.

تسمية `online` «إلكتروني» تغيّر العرض فقط.

6.2 عالج القيمة الناقصة دون إخفائها

بعض العملاء المسجلين لا تتوافر مدينتهم. بدل ترك الخلية فارغة في تقرير توضيحي، يمكن كتابة «غير مسجلة»:

```sql
SELECT
  customer_id,
  city,
  COALESCE(city, 'غير مسجلة') AS city_for_report
FROM customers
WHERE customer_id >= 14 AND customer_id <= 16
ORDER BY customer_id;
```

customer_id	city	city_for_report
14	الرياض	الرياض
15	NULL	غير مسجلة
16	الرياض	الرياض

`COALESCE` تعيد أول قيمة ليست `NULL`؛ تعرض هنا «غير مسجلة» دون تعديل المصدر. أبقِ النقص مميزًا في التحليل الجغرافي؛ العبارة ليست اسم مدينة.

6.3 تقسيم سعري: مثال لا حكم

قد يحتاج مسؤول المشتريات إلى مراجعة أسعار التمور والقهوة ضمن شريحتين مبسّطتين. سنختار 35 ريالًا حدًا **توضيحيًا** لهذا التمرين:

```sql
SELECT
  product_name,
  list_price,
  CASE
    WHEN list_price >= 35 THEN '35 ريالًا فأكثر'
    ELSE 'أقل من 35 ريالًا'
  END AS price_band
FROM products
WHERE category_id = 6
ORDER BY list_price DESC, product_id;
```

product_name	list_price	price_band
تمر عجوة 1 كجم	60.00	35 ريالًا فأكثر
هيل حب 250 جم	44.00	35 ريالًا فأكثر
قهوة عربية 500 جم	38.00	35 ريالًا فأكثر
قهوة مختصة مطحونة 250 جم	36.00	35 ريالًا فأكثر
تمر سكري 1 كجم	32.00	أقل من 35 ريالًا
تمر خلاص 1 كجم	27.00	أقل من 35 ريالًا

التصنيف مبني على سعر الرف وحد 35 ريالًا للتدريب؛ لا يثبت غلاء المنتج أو السعر المناسب للسوق.

6.4 سلّم جدولًا يستطيع غيرك قراءته

قبل نقل الناتج إلى Excel أو Power BI، راجع أربع نقاط:

1. **اسم العمود:** هل يفهم القارئ ما يمثل؟ `sales_before_returns` أدق من `sales` عندما يكون تعريفنا قبل المرتجعات.
2. **نوع القيمة:** هل الحقل تاريخ أم نص؟ هل الأسعار أرقام يمكن جمعها؟
3. **وحدة الصف:** هل يمثل كل صف منتجًا أم طلبًا أم فرعًا؟

4. **طريقة التحديث:** هل سيعطي الاستعلام الناتج نفسه من حيث التعريف عند تكراره على بيانات جديدة؟

عند التصدير إلى CSV، استخدم UTF-8 والعناوين، واستورد الأرقام والتواريخ بأنواعها. ضع «ريال» في تنسيق العرض، لا داخل القيمة الرقمية. اختر اتصال Power BI أو الاستيراد بحسب الصلاحيات.

تدريب E06 — جرّب بنفسك: غيّر التصنيف السعري في 6.3 من 35 إلى 40 ريالًا، ثم راقب أي المنتجات انتقلت بين الشريحتين. اكتب في تعليقك: «حد تدريبي، لا معيار سوقي». ما الذي يحدث لو نسيت وصف الحد بعد نقل النتيجة إلى التقرير؟

قبل أن تنتقل: كيف تحسّن العرض بـ `CASE` و `COALESCE` دون افتراض حقيقة مجهولة؟

6.5 سمِّ خطوة الحساب باستخدام CTE

CTE نتيجة وسيطة تسميها بـ `WITH` وتستخدمها في الاستعلام نفسه؛ لا تنشئ جدولًا دائمًا.

نلخص طلبات يناير وفبراير 2025 المكتملة إلى صف لكل `branch_id`، ثم نربط الملخص بالفروع.

```sql
WITH completed_by_branch AS (
  SELECT
    branch_id,
    COUNT(*) AS completed_orders
  FROM orders
  WHERE status = 'completed'
    AND order_ts >= '2025-01-01'
    AND order_ts < '2025-03-01'
  GROUP BY branch_id
)
SELECT
  b.branch_code,
  COALESCE(c.completed_orders, 0) AS completed_orders
FROM branches b
LEFT JOIN completed_by_branch c
  ON c.branch_id = b.branch_id
ORDER BY b.branch_code;
```

6.5 سمِّ خطوة الحساب باستخدام CTE

branch_code	completed_orders
AHB-01	257
DMM-01	318
JED-01	397
JED-02	327
MED-01	0
MKK-01	328
RYD-01	416
RYD-02	322

الناتج مطابق لـ5.5. يحفظ `LEFT JOIN` الفروع، وتعرض `COALESCE` العدد الغائب صفرًا. `MED-01` لم يفتتح حتى مارس؛ لا تعوّض كل مجهول بصفر تلقائيًا.

تدريب C01: حدّد موضع شرط الحالة والفترة ووحدة صف CTE، ثم طابق الناتج الكامل بـ5.5. عند تغيير الفترة، غيّرها في الاستعلامين.

CTE تنظّم القراءة والمراجعة، ولا تضمن سرعة أعلى. مرجع WITH.

07

تحقق قبل أن ترسل الرقم

سؤال الفصل:
متى يكون رقم SQL صالحًا لتقرير، ومتى يحتاج مراجعة قبل أن يصل إلى الإدارة؟

في هذا الفصل

7.1 لا تخلط الطلبات كلها بالمكتملة
7.2 من العرض إلى سبب الاختلاف
7.3 عرّف المقام وحدود النتيجة
7.4 ما الذي لا يقوله الجدول؟

قد ينجح الاستعلام ويحسب مؤشرًا غير المطلوب. تحقق من المعنى، لا من التنفيذ وحده.

ابدأ بالفترة والحالة ووحدة العد وتعريف المبلغ.

7.1 لا تخلط الطلبات كلها بالمكتملة

في 2025 يمكن أن نجمع العدد الكلي والعدد المكتمل داخل استعلام واحد:

```sql
SELECT
  COUNT(*) AS all_orders_2025,
  SUM(
    CASE WHEN status = 'completed' THEN 1 ELSE 0 END
  ) AS completed_orders_2025
FROM orders
WHERE order_ts >= '2025-01-01'
  AND order_ts < '2026-01-01';
```

all_orders_2025	completed_orders_2025
16,960	16,585

من **16,960** طلبًا في 2025، اكتمل **16,585** وأُلغي **375**. سمِّ العدد بحسب الحالات التي يشملها.

7.1 لا تخلط الطلبات كلها بالمكتملة

`CASE` تعطي 1 للمكتمل و0 لغيره، ثم تجمعها `SUM`. في الفترة الفارغة، تعيد `COUNT(*)` صفرًا و`SUM` قيمة `NULL`.

7.2 من العرض إلى سبب الاختلاف

استخدم الفحوص التي تعلمتها بدل تكرارها بلا سؤال. لكل اختلاف مسار مراجعة:

العرض	السبب المحتمل	التحقق	الإصلاح
العدد 16,960 بدل 16,585	شملت الطلبات الملغاة	قارن حالتي الطلب في 7.1	ثبّت تعريف العدد وشرط الحالة
العدد 50,079 باسم «طلبات»	تعد أسطر الطلب بعد الربط	قارن العدّين في 5.3	عدّ `order_id` المتميز للطلبات
عدد تسجيلات الولاء أقل من الطلبات	معرّفات ناقصة وطلبات متكررة للمعرّف	راجع نتائج 4.2	استخدم مقامًا يناسب السؤال
مبيعات الفرع تضاعفت بعد ربط ملف خارجي	تكرار مفتاح الربط في الملف المستورد	راجع فريدية المفتاح ووحدة الصف	أصلح علاقة الملف قبل حساب المبلغ
آخر يوم من السنة غائب	شرط النهاية يساوي بداية يوم 31 ديسمبر	راجع حدود 3.2	استخدم بداية السنة التالية غير مشمولة

7.3 عرّف المقام وحدود النتيجة

لحصة الطلبات المرتبطة بتسجيل، اقسم **10,328** على **16,585** طلبًا مكتملًا. لا تستخدم **1,535** معرّف تسجيل مقامًا لحصة الطلبات أو عددًا لأشخاص منقحين.

معرّف العميل الناقص لا يثبت أنه جديد. وصفوف الربط **50,079** متوقعة لتعدد أسطر الطلب.

قائمة المراجعة قبل التسليم

افحص	اسأل نفسك	علامة الخطر
الفترة	هل حددت البداية والنهاية بدقة؟	إسقاط آخر يوم أو إدخال سنة أخرى
الحالة	هل «مبيعات مكتملة» تستبعد الملغاة؟	عدّ الطلبات كلها دون توضيح
وحدة الصف	هل أعدّ فواتير أم أسطرًا؟	`COUNT(*)` بعد `JOIN` متعدّد
تعريف المبلغ	هل خصمت الخصومات؟ والمرتجعات؟	وصف K3 على أنه K6
القيم المفقودة	هل `NULL` تغيّر المقام؟	اختفاء سجلات دون مراجعة
المصدر	هل يمكن تتبع الرقم إلى الجداول؟	رقم من لقطة شاشة أو ملف غير معروف

ثبّت تعريف المؤشر قبل رؤية النتائج؛ تغييره لاحقًا يغيّر السؤال.

7.4 ما الذي لا يقوله الجدول؟

انخفاض المبيعات لا يثبت ضعف الإدارة؛ قد تختلف مدة التشغيل والحجم والمخزون. تفسير السبب يحتاج أدلة إضافية.

قاعدة: افصل بين **النتيجة المحسوبة** و**تفسيرها** و**التوصية**. لا تجعل احتمالًا يبدو كأنه حقيقة لمجرد أن الاستعلام نجح.

> **تدريب E07 — جرّب بنفسك:** غيّر الفترة في فحص 7.1 إلى 2024، ثم اكتب في سطر واحد لماذا لا يجوز مقارنة عدد طلبات 2024 و2025 دون معرفة ما تغير في عدد الفروع أو مدة تشغيلها.

> **قبل أن تنتقل:** اكتب مصدر الرقم وفترته وشروطه ووحدته وما يستبعده.

08

مشروع مصغّر: أداء الطلبات والمبيعات

سؤال المشروع:
كيف تبني ملخصًا واضحًا لأداء الفروع في 2025، ثم تتحقق منه قبل إرساله؟

في هذا الفصل

8.1 الخطوة الأولى: اعرف الفروع التي تقارنها
8.2 الخطوة الثانية: العدد أولًا
8.3 الخطوة الثالثة: أضف مبلغًا بتعريف صريح
8.4 الخطوة الرابعة: جدول واحد للتسليم
8.5 تحقق قبل إرسال الملف
8.6 ماذا حققت؟

مشروع مصغّر: أداء الطلبات والمبيعات

المهمة: جدول لكل فرع في 2025 يعرض اسمه، والطلبات المكتملة، والمبيعات بعد خصومات السطور وقبل المرتجعات، ومتوسطها لكل طلب مكتمل.

سلّم الجدول وتعريف مؤشرات مختصرًا وملاحظة عن حدود المقارنة.

8.1 الخطوة الأولى: اعرف الفروع التي تقارنها

قبل حساب أي شيء، اطلع على الفروع وتاريخ افتتاحها:

```sql
SELECT
  branch_code,
  city,
  opened_on
FROM branches
WHERE opened_on < '2026-01-01'
ORDER BY branch_code;
```

النتيجة الكاملة:

branch_code	city	opened_on
AHB-01	أبها	2023-06-01
DMM-01	الدمام	2020-03-01
JED-01	جدة	2018-11-01
JED-02	جدة	2022-02-01
MED-01	المدينة المنورة	2025-03-01
MKK-01	مكة المكرمة	2020-10-01
RYD-01	الرياض	2019-05-01
RYD-02	الرياض	2021-09-15

افتتح `MED-01` في مارس 2025؛ أضف ملاحظة عن مدة تشغيله عند المقارنة السنوية.

8.2 الخطوة الثانية: العدد أولًا

شغّل 5.1 لطلبات 2025 المكتملة. طابق مجموع الفروع الثمانية مع **16,585** في 4.1.

8.3 الخطوة الثالثة: أضف مبلغًا بتعريف صريح

شغّل 5.4: الكمية × سعر البيع الفعلي − خصم السطر. أعلى مبلغ **284,948.58** لفرع `RYD-01`.

> استخدم عنوان المؤشر «المبيعات بعد خصومات السطور وقبل المرتجعات لعام 2025» (`K3`)، وأضف متوسط الطلب من 5.4.

8.4 الخطوة الرابعة: جدول واحد للتسليم

ادمج العدد والمبلغ. بعد ربط الأسطر، تعدّ `COUNT(DISTINCT o.order_id)` الطلبات مرة واحدة:

```sql
SELECT
  b.branch_name,
  COUNT(DISTINCT o.order_id) AS completed_orders,
  ROUND(
    SUM(oi.quantity * oi.unit_price - oi.discount_amount),
    2
  ) AS sales_before_returns,
  ROUND(
    SUM(oi.quantity * oi.unit_price - oi.discount_amount)
      / COUNT(DISTINCT o.order_id),
    2
  ) AS avg_before_returns_per_order
FROM orders o
JOIN order_items oi ON oi.order_id = o.order_id
JOIN branches b ON b.branch_id = o.branch_id
WHERE o.status = 'completed'
  AND o.order_ts >= '2025-01-01'
  AND o.order_ts < '2026-01-01'
GROUP BY b.branch_id, b.branch_name
ORDER BY sales_before_returns DESC, b.branch_name;
```

8.4 الخطوة الرابعة: جدول واحد للتسليم

الجدول النهائي للقراءة:

branch_name	completed_orders	sales_before_returns	avg_before_returns_per_order
الرياض - العليا	2,778	284,948.58	102.57
جدة - الروضة	2,661	270,755.79	101.75
الرياض - النرجس	2,158	220,883.79	102.36
مكة - العزيزية	2,022	207,340.59	102.54
الدمام - الشاطئ	1,968	206,314.77	104.83
جدة - أبحر	1,934	195,104.09	100.88
أبها - المنسك	1,605	168,438.85	104.95
المدينة - قباء	1,459	148,632.21	101.87

المتوسط = المبلغ قبل المرتجعات ÷ الطلبات المكتملة؛ لا يمثل متوسط السطر أو الصافي بعد المرتجعات.

كيف تقرأ التقرير؟

اسم الفرع يميز فروع المدينة ويجعل التقرير أوضح.

عدد الطلبات يقيس حجم النشاط، والمبلغ يقيس قيمة المبيعات.

راعِ قِصر مدة تشغيل `MED-01` في المقارنة.

الجدول لا يفسر أسباب الفروق أو يقيس الربحية.

8.5 تحقق قبل إرسال الملف

نفّذ المراجعة التالية على الناتج:

8.5 تحقق قبل إرسال الملف

1. **الصفوف:** صف لكل فرع؛ راجع الربط والتجميع عند التكرار.
2. **الطلبات:** مجموع الفروع **16,585**.
3. **المبلغ:** مجموع الفروع **1,702,418.67 ريال**؛ طابقه بالحساب أدناه.
4. **الفترة:** بداية 2025 مشمولة، وبداية 2026 مستبعدة.
5. **العنوان:** وضّح أن المبلغ قبل المرتجعات.
6. **المقارنة:** أرفق تاريخ افتتاح الفرع الجديد.

حساب مستقل لمطابقة الإجمالي

```sql
SELECT
  COUNT(DISTINCT o.order_id) AS completed_orders,
  ROUND(
    SUM(oi.quantity * oi.unit_price - oi.discount_amount),
    2
  ) AS sales_before_returns
FROM orders o
JOIN order_items oi ON oi.order_id = o.order_id
WHERE o.status = 'completed'
  AND o.order_ts >= '2025-01-01'
  AND o.order_ts < '2026-01-01';
```

completed_orders	sales_before_returns
16,585	1,702,418.67

الفحص يطابق الإجمالي دون جدول الفروع؛ لا يثبت صحة المصدر. متوسط الشركة = إجمالي المبلغ ÷ إجمالي الطلبات، وليس متوسط متوسطات الفروع.

ملخص مرفق للمدير

«النتيجة لطلبات 2025 المكتملة، بعد خصومات السطور وقبل المرتجعات، ودون الضريبة. فرع المدينة افتتح في مارس؛ تُراعى مدة تشغيله عند المقارنة. الجدول لا يقيس الربحية أو يفسر أسباب الفروق».

ارفق التعريف بالجدول. الربط الداخلي مناسب هنا لأن لكل فرع طلبات مكتملة. لإظهار فروع بلا طلبات في فترة أخرى، استخدم `LEFT JOIN` وعالج المقام صفرًا قبل حساب المتوسط.

8.6 ماذا حققت؟

أصبحت تستطيع تحديد السؤال، وحساب المؤشرات، والتحقق من الناتج وتسليمه بتعريف واضح.

> **تدريب E08 — التحدي النهائي:** طبّق على فرع `RYD-01` (المعرّف `1`) لتقارن بحل قابل لإعادة التنفيذ. اعرض اسمه وعدد طلباته المكتملة ومبلغ سطورها قبل المرتجعات ومتوسط الطلب لعام 2025. ثم يمكنك اختيار فرع آخر. اكتب تحته ثلاثة أسطر: ماذا حسبت؟ ماذا لم تحسب؟ وماذا تحتاج قبل مقارنة الفرع بغيره؟

المرجع السريع — Cheat Sheet

الأداة	الوظيفة	أهم سؤال قبل استخدامها
`ORDER BY`	رتّب النتائج	الأعلى أو الأقل بحسب أي عمود؟
`ROUND`	قرّب قيمة للعرض	هل أبقيت الحساب قبل التقريب؟
`COUNT`	عدّ الصفوف أو القيم	طلبات أم أسطر أم معرّفات تسجيل؟
`SUM`	اجمع قيمة	هل المبلغ بعد الخصومات؟ والمرتجعات؟
`GROUP BY`	لخّص بحسب مجموعة	ماذا يمثل صف النتيجة الجديد؟
`HAVING`	صفِّ نتيجة المجموعات	هل الشرط على المجموعة لا على الصف؟
`JOIN`	اربط البيانات المتطابقة	ما المفتاح والعلاقة؟
`LEFT JOIN`	احتفظ بصفوف الجانب الأيسر	هل يجب إظهار الصف بلا مطابق؟
`CASE`	صنّف بحسب شرط	هل المعايير موثقة ومناسبة؟
`COALESCE`	أظهر بديلًا للقيمة `NULL`	هل البديل تسمية عرض أم قيمة فعلية؟

حلول التمارين ومؤشرات التحقق

تخص هذه الحلول التمارين E05–E08. حافظنا على أرقامها لتبقى متصلة بمسار SQL 01. قارن الكود والنتيجة والتفسير؛ إجابة C01 موضحة بجانب المثال في 6.5.

حل E05 — اسم الفرع دون تغيير عدد الطلبات

```sql
SELECT
  b.branch_name,
  b.city,
  COUNT(*) AS completed_orders
FROM orders o
JOIN branches b ON b.branch_id = o.branch_id
WHERE o.status = 'completed'
  AND o.order_ts >= '2025-01-01'
  AND o.order_ts < '2026-01-01'
GROUP BY b.branch_id, b.branch_name, b.city
ORDER BY completed_orders DESC, b.branch_id;
```

branch_name	city	completed_orders
الرياض - العليا	الرياض	2,778
جدة - الروضة	جدة	2,661
الرياض - النرجس	الرياض	2,158
مكة - العزيزية	مكة المكرمة	2,022
الدمام - الشاطئ	الدمام	1,968
جدة - أبحر	جدة	1,934
أبها - المنسك	أبها	1,605
المدينة - قباء	المدينة المنورة	1,459

حل E05 — اسم الفرع دون تغيير عدد الطلبات

قارن الأعداد بمعرّفات branch_id الصفوف 4.3: لم يتغير عدد الطلبات؛ كل طلب يطابق فرعًا واحدًا. يجمع بحسب الهوية، ولو تشابه اسم فرعين. في **ملف خارجي** فيه صفان للمعرّف نفسه، سيطابق كل طلب صفين، وقد يتضاعف العد والمبلغ. أصلح فريدية الملف أو العلاقة؛ لا تستعمل DISTINCT لإخفاء المشكلة. القاعدة الأصلية تفرض مفتاحًا أساسيًا يمنع هذا التكرار.

حل E06 — تغيير الحد إلى 40 ريالًا

```sql
SELECT
  product_name,
  list_price,
  CASE
    WHEN list_price >= 40 THEN '40 ريالًا فأكثر'
    ELSE 'أقل من 40 ريالًا'
  END AS price_band
FROM products
WHERE category_id = 6
ORDER BY list_price DESC, product_id;
```

product_name	list_price	price_band
تمر عجوة 1 كجم	60.00	40 ريالًا فأكثر
هيل حب 250 جم	44.00	40 ريالًا فأكثر
قهوة عربية 500 جم	38.00	أقل من 40 ريالًا
قهوة مختصة مطحونة 250 جم	36.00	أقل من 40 ريالًا
تمر سكري 1 كجم	32.00	أقل من 40 ريالًا
تمر خلاص 1 كجم	27.00	أقل من 40 ريالًا

حل E06 — تغيير الحد إلى 40 ريالًا

انتقلت القهوة العربية (38) والقهوة المختصة (36) إلى الشريحة الأقل من الحد. بقي التمر العجوة (60) والهيل (44) في الشريحة الأعلى؛ السكري (32) والخلاص (27) دون الحد في الحالتين. أرفق عبارة **«حد تدريبي، لا معيار سوقي»** حتى لا يقرأ المتلقي التصنيف حكمًا على الأسعار.

حل E07 — مراجعة 2024

```sql
SELECT
  COUNT(*) AS all_orders_2024,
  SUM(
    CASE WHEN status = 'completed' THEN 1 ELSE 0 END
  ) AS completed_orders_2024
FROM orders
WHERE order_ts >= '2024-01-01'
  AND order_ts < '2025-01-01';
```

all_orders_2024	completed_orders_2024
14,514	14,224

الفرق **290** طلبًا ملغًى. مقارنة الإجماليين السنويين تتأثر بإضافة فرع المدينة في مارس 2025 ومدة تشغيله؛ نحتاج فروعًا وفترات قابلة للمقارنة قبل تفسير الفرق نموًا في الأداء.

حل E08 — تقرير فرع الرياض العليا

```sql
SELECT
  b.branch_name,
  COUNT(DISTINCT o.order_id) AS completed_orders,
  ROUND(
    SUM(oi.quantity * oi.unit_price - oi.discount_amount),
    2
  ) AS sales_before_returns,
  ROUND(
    SUM(oi.quantity * oi.unit_price - oi.discount_amount)
      / COUNT(DISTINCT o.order_id),
    2
  ) AS avg_before_returns_per_order
FROM orders o
JOIN order_items oi ON oi.order_id = o.order_id
JOIN branches b ON b.branch_id = o.branch_id
WHERE o.status = 'completed'
  AND o.branch_id = 1
  AND o.order_ts >= '2025-01-01'
  AND o.order_ts < '2026-01-01'
GROUP BY b.branch_id, b.branch_name;
```

branch_name	completed_orders	sales_before_returns	avg_before_returns_per_order
الرياض - العليا	2,778	284,948.58	102.57

1. حسبت **2,778** طلبًا مكتملًا في 2025، ومبلغ **284,948.58 ريال** بعد خصومات السطور وقبل المرتجعات، ومتوسط **102.57 ريال لكل طلب**.

2. لم أخصم المرتجعات ولم أحسب الربحية أو الضريبة أو أسباب الأداء.

3. قبل المقارنة أحتاج مدة تشغيل الفروع وحجمها وسياقًا للمؤشر، وأحسب متوسط الشركة من إجمالي المبلغ وإجمالي الطلبات، لا من متوسط متوسطات الفروع.

مؤشرات تحقق جامعة: في المشروع ثمانية صفوف، ومجموع الطلبات 16,585، ومجموع المبلغ 1,702,418.67. لا يستنتج عدد الأشخاص من معرّفات تسجيل الولاء، ولا يساوي عدد الأسطر عدد الطلبات.

المصطلح	المعنى	مثال
Primary Key / مفتاح أساسي	حقل أو مجموعة حقول تعرّف الصف تعريفًا فريدًا.	`orders.order_id` يعرّف الطلب.
Foreign Key / مفتاح أجنبي	حقل يشير إلى صف في جدول آخر.	`orders.branch_id` يشير إلى الفرع.
Grain / وحدة الصف	ما يمثله الصف الواحد في الجدول أو النتيجة.	بعد التجميع: صف واحد لكل فرع.
NULL / قيمة غير معروفة	قيمة مفقودة أو غير معروفة وفق تعريف البيانات؛ ليست صفرًا.	`city IS NULL` يختار مدينة غير مسجلة.
KPI / مؤشر أداء	رقم له تعريف ثابت ويرتبط بسؤال أو قرار.	عدد الطلبات المكتملة خلال 2025.
Sales before returns / مبيعات قبل المرتجعات	مبالغ الطلبات المكتملة بعد خصومات السطور وقبل خصم المرتجعات.	المؤشر K3 في بيانات التدريب.

المصادر وتنبيهات الاستخدام

بيانات التدريب: الأمثلة من قاعدة «متاجر واحة» الافتراضية، ولا تمثل بيانات شركة حقيقية. تتضمن الحزمة `schema.sql` وCSV و`DATA_DICTIONARY.md` لتعريف الحقول والمؤشرات.

مراجع PostgreSQL: SELECT · الدوال التجميعية · الدوال الشرطية · WITH / CTE.

قبل استخدام الأمثلة في عملك: الصيغ مكتوبة لـPostgreSQL وقد تحتاج تعديلًا لمحرك آخر. تحقق من تعريف الحقول والمؤشرات وصلاحيات الوصول وخصوصية البيانات. نجاح الاستعلام على بيانات التدريب لا يثبت صحة تقرير العمل؛ إدارة القواعد والتعديل والأداء المتقدم خارج نطاق الجزأين.

© 2026 **عبدالرحمن الأكلبي. جميع الحقوق محفوظة.** يُسمح بمشاركة الدليل كاملًا مجانًا دون تعديل، مع إبقاء الاسم وروابط المصدر. لا يُسمح ببيعه أو نسبته إلى غير مُعدّه أو إعادة نشر محتواه المعدّل دون إذن. تبقى حقوق المصادر والخطوط والشعارات لأصحابها.

الموقع **aalaklabi.com** هو المصدر الرسمي لهذا الدليل ونسخه المحدّثة. عند مشاركة الدليل أو الاقتباس منه، يُرجى ذكر اسم الدليل واسم مُعدّه وإرفاق رابط الموقع.

شكرًا لقراءتك

أصبحت تستطيع ربط الجداول وتجهيز مؤشرات واضحة ومراجعة نتائجها.
عد إلى قائمة التحقق قبل تسليم أي تقرير، وطبّق المشروع على سؤال محدد من عملك.



عن الكاتب

عبدالرحمن الأكلبي

أقدّم أدلة عملية تبسّط الأدوات والمهارات التقنية، وتساعد على فهمها واستخدامها في الدراسة والعمل.

aalaklabi.com
linkedin.com/in/aalaklabi