المحاضرة #٧ أنظمة المعلومات

١ مقدمة

في المحاضرة الثانية تعرفنا على البيانات (Data) وعرفناها على إنها قيم غير منظمة. كما تعرفنا على المعلومات (Information) وعرفناها على إنها البيانات التي تم تنظيمها أو معالجتها بشكل ما لاستخدامها في حل مسألة ما. في هذه المحاضرة سوف نتعرف على نظم المعلومات (Information System) وهي برمجيات تساعدنا على تنظيم وتحليل البيانات إي إنها ببساطة تساعدنا على استخلاص المعلومات من البيانات. لتحقيق هذا الهدف تعمل نظم المعلومات على تصنيف وتجميع وتحويل البيانات بطرق مختلفة لجعلها ذات معنى. في الغالب أدارة المعلومات ليست مهمة فردية. عادة ما تتطلب تعاون جماعي والعمل مع خبراء البيانات لرؤية أهميتها من مختلف الزوايا. في هذه المحاضرة سوف نتعرف على اهم ثلاث تطبيقات متعددة الأغراض لأنظمة المعلومات وهي الجداول الحسابية الإلكترونية (Electronic Spreadsheets) وأنظمة أدارة قواعد البيانات (Database Management Systems) والتجارة الإلكترونية (E-Commerce).

٢ جداول البيانات (Spreadsheets)

تتوفر العديد من برامج جداول البيانات في الوقت الحالي مثل برنامَج اكسل (Excel) من مايكروسوفت (Microsoft) او سبريدشيت (Spreadsheet) من جوجل (Google) او كالك (Calc) من لايبراوفيس (LibreOffice). قد توجد اختلافات بسيطة بين هذه البرمجيات من حيث الإمكانات وصيغة بعض العبارات إلا إن جميعها يتبنى نفس مبادئ العمل. أن جداول البيانات (Spreadsheet) هي تطبيقات برمجية تساعدنا علي تنظيم وتحليل البيانات بواسطة شبكة من الخلايا (Cells). قد تحتوي الخلية على بيانات أو معادلة لحساب قيمة. البيانات المخزونة في الخلية قد تكون نصوص أو إعداد أو قيم خاصة مثل التاريخ.

	A	B	C	D
1				
2				
3				
4				
5				

شكل ١: جزء من جدول بيانات

مصدر الشكل كتاب Computer Science Illuminated by Nell B. Dale

كما هو ظاهر في الشكل ١ تعرف الخلايا بواسطة الأعمدة والصفوف. غالبا ما تحدد الأعمدة بواسطة الأحرف وتحدد الصفوف بواسطة الأرقام. وبذلك عنوان الخلية يكون كالتالي C4 أو B22 أو Y1. بعد ٢٦ عمود يستخدم حرفان لتحديد العمود مثلا الخلية AA5. عادة ما تحتوي معظم برامج جداول البيانات على عدد صفوف كبير وكافي. كذلك تحتوي هذه البرامج على أمكانية أضافه ورقة (sheet) تحتوي على جدول مشابه للورقة الأولى.

تعتبر جداول البيانات مفيدة في حالات كثيرة. غالبا ما يتم تصميم هذه البرمجيات لتدير الآلاف من بيانات القيم والحسابات. لنأخذ مثلا صغيرًا لتوضيح مبادئ جداول البيانات الأساسية. لنفترض أننا جمعنا بيانات عن عدد الطلاب الذين أتوا للحصول على المساعدة من مجموعة من المدرسين على مدى عدة أسابيع. لقد تابعنا عدد الطلاب الذين ذهبوا إلى كل من المدرسين الثلاثة هال (Hal) وإيمي (Amy) وفرانك (Frank) كل أسبوع لمدة خمسة أسابيع. مع إجراء بعض التحليلات الأساسية على تلك البيانات يتكون لنا جدول البيانات الموضح في الشكل ٢.

يحتوي جدول البيانات هذا على مجموعة من البيانات تتطلب تحليل أضافه الى أمور أخرى. مثلا تحتوي الخلية C4 على عدد الطلاب الذين يقوم المدرس هال (Hal) بمساعدتهم في الأسبوع الأول. أعمدة البيانات من C4 إلى C8 تحتوي على عدد الطلاب الذين ساعدهم هال (Hal) في الأسابيع الخمسة التي تم تجميع البيانات فيها. وبشكل مشابه فان بيانات ايمي (Amy) مخزونه في الأعمدة من D4 إلى D8 وبيانات فرانك (Frank) مخزونة من العمود E4 إلى E8. يمكن النظر إلى صفوف البيانات بدلا من الأعمدة. فمثلا كل صف يمثل عدد الطلبة الذين تمت مساعدتهم من قبل المدرسين في إي أسبوع.

في الخلايا C9 وD9 وE9 يحسب جدول البيانات العدد الكلي للطلاب الذين تمت مساعدتهم من قبل كل مدرس لجميع الأسابيع الخمسة. في الخلايا C10 وD10 وE10 يحسب جدول البيانات معدل للطلاب الذين تمت مساعدتهم من قبل كل مدرس لكل أسبوع. وبشكل مشابه فان العدد الكلي للطلاب الذين تمت مساعدتهم لكل أسبوع (من جميع المدرسين) ظاهرة في أعمدة الخلايا من F4 الى F8. معدل الطلبة الذين تمت مساعدتهم لكل أسبوع ظاهر في الخلايا من G4 إلى G8.

يقوم جدول البيانات بحساب إحصاءات أجمالية أخرى أضافه إلى حساب المعدل و المجموع لكل مدرس ولكل أسبوع. مثلا الخلية F9 تظهر العدد الكلي للطلاب الذين تمت مساعدتهم لكل الأسابيع

والمعدل لكل أسبوع (لكل المدرسين) يظهر في الخلية F10 والمعدل لكل مدرس (لكل الأسابيع) يظهر في الخلية G9. أخيرًا فان معدل الطلبة الذين تمت مساعدتهم من اي مدرس في اي أسبوع يظهر في الخلية G10.

	A	B	C	D	E	F	G	H
1								
2				Tutor				
3			Hal	Amy	Frank	Total	Avg	
4		1	12	10	13	35	11.67	
5		2	14	16	16	46	15.33	
6	Week	3	10	18	13	41	13.67	
7		4	8	21	18	47	15.67	
8		5	15	18	12	45	15.00	
9		Total	59	83	72	214	71.33	
10		Avg	11.80	16.60	14.40	42.80	14.27	
11								
12								

شكل ٢: جدول بيانات يحتوي بيانات وحسابات

مصدر الشكل كتاب Computer Science Illuminated by Nell B. Dale

إن البيانات المخزونة في العمودين A وB والصفين ٢ و٣ تستخدم ببساطة لتوضيح ما تمثله القيم الموجودة في بقية الجدول. تهدف هذه العناوين لتوضيح البيانات ولا عِلاقة لها بالحسابات الرياضية. نلاحظ ان بعض الخلايا في الشكل ٢ تظهر بألوان مختلفة. معظم برامج جداول البيانات تسمح للمستخدم بالسيطرة على شكل وصيغة البيانات في الخلايا بطرق مختلفة. يستطيع المستخدم تحديد نوع الخط وشكلة ولونه أضافه إلى اتجاه (Alignment) البيانات داخل الخلية (توسيط مثلا). في حالة قيم الأعداد الحقيقية كما هو في المعدلات المحسوبة في مثالنا, يستطيع المستخدم تحديد عدد المراتب بعد الفاصلة المنقوطة المطلوب عرضها. كذلك يمكن أظهار و أخفاء شبكة الخطوط المنقطة بين الخلايا وتغيير لون خلفية الجدول في معظم برامج جداول البيانات.

٢-١ صيغ جداول البيانات (Spreadsheet Formulas)

في مثالنا السابق، رأينا العديد من العمليات الحسابية التي أعطتنا نظرة عامة عن مساعدة المدرسين للطلبة. يعتبر إعداد هذه الحسابات سهل نسبيا. قد تقول إن أجراء هذه الحسابات يدويا بمساعدة الآله

الحاسبة لا يتطلب وقتاً طويلا. وقد يكون هذا صحيحاً ولكن ميزة جداول البيانات تكمن في سهولة تعديلها وتوسيعها بسهولة. إذا أعددنا جدول البيانات بشكل صحيح، فيمكننا إضافة المدرسين أو إزالتهم ، أو إضافة أسابيع إضافية من البيانات ، أو تغيير أي من البيانات التي قمنا بتخزينها بالفعل وسيتم تحديث الحسابات المقابلة تلقائيًا. على سبيل المثال ، على الرغم من قيامنا بإعداد جدول بيانات المدرس لاستخدام بيانات ثلاثة مدرسين ، يمكن توسيع نفس جدول البيانات للتعامل مع مئات المدرسين. بدلاً من خمسة أسابيع من البيانات ، يمكننا بنفس السهولة معالجة قيمة عام كامل.

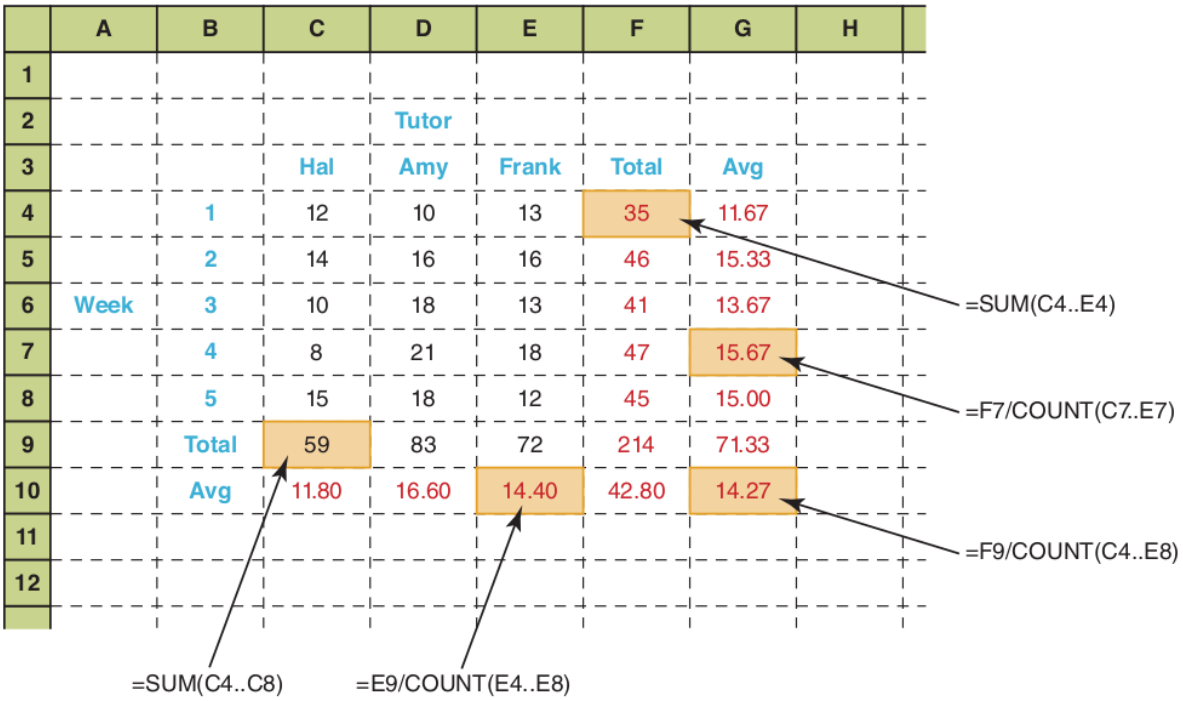
تأتي قوة جداول البيانات من الصيغ التي يمكننا إنشاؤها وتخزينها في الخلايا. تم حساب جميع الإجماليات والمعدلات في المثال في الشكل ٢ باستخدام الصيغ (Formulas). عندما يتم تخزين صيغة في خلية، يتم عرض نتيجة الصيغة في الخلية. لذلك، عندما ننظر إلى القيم الموجودة في جدول البيانات، يكون من الصعب أحيانًا معرفة ما إذا كانت البيانات المعروضة في خلية معينة قد تم إدخالها مباشرةً أو تم حسابها بواسطة صيغة أساسية.

يوضح الشكل ٣ نفس جدول البيانات في الشكل ٢، لكنه يشير إلى الصيغ الكامنة وراء بعض الخلايا. تبدأ الصيغ في أمثلتنا (كما هو الحال في العديد من برامج جداول البيانات) بعلامة يساوي (=). هذه هي الطريقة التي يعرف بها جدول البيانات الخلايا التي تحتوي على الصيغ التي يجب حسابها. يتم استخدام القيم المخزنة في الخلايا المرجعية (اي الخلايا التي تعتمد عليها الصيغة لحساب النتيجة) لحساب نتيجة الصيغة . كلما تغيرت قيم جدول البيانات يتم إعادة حساب نتيجة الصيغ الموجودة في جدول البيانات إي إن النتائج تكون حديثة دائمًا. يعتبر جدول البيانات ديناميكيًا - فهو يستجيب للتغييرات على الفور. إذا قمنا بتغيير عدد الطلاب الذين درسهم فرانك (Frank) في الأسبوع الثاني ، فسيتم إعادة حساب الإجماليات والمعدلات التي تستخدم هذه القيمة على الفور لتعكس البيانات المنقحة.

يمكن للصيغ الاستفادة من العمليات الحسابية الأساسية باستخدام الرموز القياسية (+ و ٢ و * و /). يمكنهم أيضًا الاستفادة من **دوال جداول البيانات (Spreadsheet Functions)** المضمنة في البرنامج. في مثال المدرس، تستخدم الصيغة في الخلية C9 الدالة SUM لحساب مجموع القيم في الخلايا C4 و C5 و C6 و C7 و C8.

نظرًا لأن ألدوال تعمل غالبًا على مجموعة من الخلايا المتجاورة ، فإن جداول البيانات توفر طريقة

مناسبة لتحديد نطاق من الخلايا.



شكل ٣: الصيغ المستخدمة في خلايا الجدول

مصدر الشكل كتاب Computer Science Illuminated by Nell B. Dale

يتم تحديد النطاق بنقطتين (..) بين أول وأخر خلية في النطاق. يمكن أن يحدد النطاق مجموعة من الخلايا على طول الصف مثل C4..E4 ، أو يمكنه تحديد مجموعة من الخلايا على طول العمود مثل C4..C8. يمكن أن يحدد النطاق أيضًا كتلة مستطيلة من الخلايا. على سبيل المثال ، يتضمن النطاق C4..E8 الخلايا من C4 إلى C8 ومن D4 إلى D8 ومن E4 إلى E8.

تستخدم العديد من الصيغ الموضحة في الشكل ٣ الدالة COUNT والتي تحسب عدد الخلايا غير الفارغة في النطاق المحدد. على سبيل المثال ، تقسم الصيغة في الخلية G7 القيمة الموجودة في الخلية F7 على عدد الخلايا في النطاق C7..E7 ، وهو ٣. علية يمكن كتابة الصيغة في الخلية G7 على النحو التالي:

=SUM(C7..E7)/3

اعتماداً على قيم جدول البيانات الحالية ستحسب هذه الصيغة النتيجة نفسها، لكن هذه الصيغة ليست بجودة الصيغة الأصلية "F7/COUNT(C7..E7)" والتي تستخدم الدالة COUNT لسببين. أولاً، تم بالفعل حساب مجموع القيم في C7 إلى E7 (وتخزينه في F7)، لذلك ليست هناك حاجة لإعادة حسابها من جديد. لان اي تغيير في البيانات سيؤثر على قيمة F7، وبالتالي يغير قيمة G7

أيضًا. تأخذ جداول البيانات كل هذه العلاقات في الاعتبار.

ثانيًا (وهو الأهم)، من الأفضل دائمًا تجنب استخدام ثابت في صيغة ما لم يكن مناسبًا على وجه التحديد. في هذه الحالة، فإن استخدام القيمة ٣ كعدد محدد مسبقًا للمدرسين يحد من قدرتنا على إضافة أو حذف المدرسين بسهولة من تحليلنا. تستجيب صيغ جداول البيانات لعمليات الإدراج والحذف تمامًا كما تفعل مع التغييرات في البيانات الأولية نفسها. إذا أدرجنا عمودًا لمدرس آخر، فإن النطاقات في الصيغ الأصلية في العمودين F و G (والتي ستنتقل إلى العمودين G و H بسبب الإدراج) ستتغير تلقائيًا لتعكس الإدراج. على سبيل المثال، إذا تم إدراج عمود مدرس جديد، فسيتم نقل الصيغة الموجودة في الخلية F4 إلى الخلية G4 وستصبح:

=SUM(C4..F4)

أي أن نطاق الخلايا سيزداد ليشمل البيانات المدرجة حديثًا. وبالمثل، فإن النطاقات المستخدمة بواسطة الدالة COUNT في ألدوال الأخرى قد تتغير مما ينتج عنه معدل جديد وصحيح. إذا استخدمنا الثابت ٣ في صيغة الخلية G7 ، فسيكون الحساب غير صحيح بعد إدراج العمود الجديد.

عادةً ما يوفر برنامج جداول البيانات عددًا كبيرًا من ألدوال التي يمكننا استخدامها في الصيغ. بعض هذه ألدوال يقوم بإجراء حسابات رياضية أو إحصائية أو حسابات مالية شائعة أو عمليات خاصة على نص أو تواريخ. يسمح البعض الآخر للمستخدم بإقامة علاقات منطقية بين الخلايا. تظهر أمثلة لبعض دوال جداول البيانات الشائعة في الشكل ٤. توفر معظم برامج جداول البيانات عشرات ألدوال التي يستطيع المستخدم أدماجها في الصيغ.

جانب ديناميكي آخر لجداول البيانات هو القدرة على نسخ القيم أو الصيغ عبر صف أو أسفل عمود. عند نسخ الصيغ ، يتم الحفاظ على العلاقات بين الخلايا. نتيجة لذلك ، يصبح من السهل إعداد مجموعة كاملة من الحسابات المتشابهة. على سبيل المثال ، لإدخال حساب المجموع في مثال المدرس الخاص بنا أسفل العمود من الخلية F4 إلى الخلية F8 ، كان علينا ببساطة إدخال الصيغة في الخلية F4 ، ثم نسخ هذه الصيغة أسفل العمود. أثناء نسخ الصيغة ، يتم تحديث المراجع إلى الخلايا تلقائيًا لتعكس الصف الذي توجد به الصيغة الجديدة. بالنسبة لمثالنا الصغير الذي يتتبع خمسة أسابيع ، لم توفر إمكانية النسخ الكثير من الجهد. لكن تخيل أننا كنا نتتبع هذه البيانات لمدة عام كامل وكان لدينا ٥٢ صيغة جمع لإنشاءها. إن ميزة النسخ هذه في جدول البيانات تجعلنا قادرين على إعداد كل العمود بعملية

واحدة فقط.

Function	Computes
SUM(val1, val2, ...) SUM(range)	Sum of the specified set of values
COUNT(val1, val2, ...) COUNT(range)	Count of the number of cells that contain values
MAX(val1, val2, ...) MAX(range)	Largest value from the specified set of values
SIN(angle)	The sine of the specified angle
PI()	The value of pi
STDEV(val1, val2, ...) STDEV(range)	The standard deviation from the specified sample values
TODAY()	Today's date
LEFT(text, num_chars)	The leftmost characters from the specified text
IF(test, true_val, false_val)	If the test is true, it returns the true_val; otherwise, it returns the false_val
ISBLANK (value)	Returns true if the specified value refers to an empty cell

شكل ٤: بعض دوال جداول البيانات الشائعة

مصدر الشكل كتاب Computer Science Illuminated by Nell B. Dale

٢-٢ التحليل باستخدام جداول البيانات (Spreadsheet Analysis)

أحد الأسباب التي تجعل جداول البيانات مفيدة للغاية هو قدرتها على التكيف. يحدد مستخدم جدول البيانات ما تمثله البيانات وكيفية ارتباطها بالبيانات الأخرى. لذلك ، يمكن تطبيق تحليل جدول البيانات على أي مجال موضوع تقريبًا. على سبيل المثال يمكن ان نستخدم جدول البيانات لأداء المهام التالية:

- تتبع المبيعات
- تحليل الإحصائيات الرياضية
- خزن وأداره درجات الطالب
- الاحتفاظ بسجل صيانة السيارة
- تسجيل وتلخص نفقات السفر
- تتبع أنشطة المشروع والجداول الزمنية
- التخطيط لشراء الأسهم

ان قائمة التطبيقات المحتملة تكاد لا تنتهي. بشكل عام لدى قطاع الأعمال عدد كبير من الحالات

المحددة التي تكون فيها حسابات جداول البيانات ضرورية. يجعلك تتساءل كيف تعاملنا بدونهم.
أن الطبيعة الديناميكية لجداول البيانات تجعلها مفيدة للغاية. إذا قمنا بإعداد صيغ جدول البيانات بشكل صحيح ، فسيتم أخذ التغييرات والإضافات والحذف إلى البيانات في الاعتبار تلقائيًا من خلال الحسابات المناسبة.

توفر الطبيعة الديناميكية لجداول البيانات قدرة قوية على إجراء **تحليل ماذا لو**. يمكننا إعداد جداول بيانات تأخذ في الاعتبار افتراضات معينة ، ثم تتحدى تلك الافتراضات عن طريق تغيير القيم المناسبة. كمثال، لنفترض أننا نقوم بإعداد جدول بيانات لتقدير التكاليف والأرباح المحتملة لندوة نعتزم عقدها. يمكننا إدخال قيم لعدد الحضور وأسعار التذاكر وتكلفة المواد واستئجار الغرف والبيانات الأخرى التي تؤثر على النتائج النهائية. ثم يمكننا أن نسأل أنفسنا بعض أسئلة **ماذا لو (What-if)** لنرى كيف يتغير السيناريو الخاص بنا مع تغير الموقف:

- ماذا لو انخفض عدد الحاضرين بنسبة ١٠٪؟
- ماذا لو قمنا بزيادة سعر التذكرة بمقدار ٥ دولارات؟
- ماذا لو تمكنا من تقليل تكلفة المواد إلى النصف؟

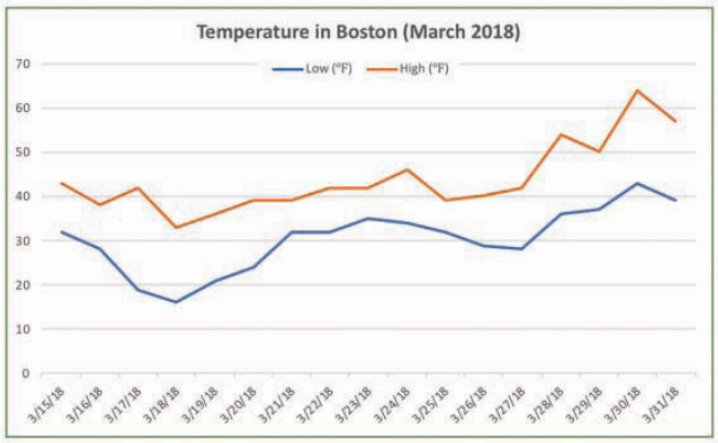
بينما نطرح هذه الأسئلة ، نقوم بتغيير البيانات وفقًا لذلك. إذا أعددنا العلاقات بين جميع الصيغ بشكل صحيح ، فسيظهر لنا كل تغيير على الفور كيف يؤثر على البيانات الأخرى.
قام محللو الأعمال بصياغة هذه العملية بطرق مختلفة، وأصبحت جداول البيانات أداة أساسية في عملهم اليومي. أصبح تحليل التكلفة والفوائد (Cost-benefit analysis) وحسابات التعادل (break-even calculations) وتقديرات المبيعات المتوقعة كلها مسألة تنظيم بيانات جدول البيانات والصيغ لأخذ العلاقات المناسبة في الاعتبار.

٢-٣ التمثيل البياني باستخدام جداول البيانات (Spreadsheet Visualizations)

يقال إن الصورة تساوي ألف كلمة. قد يكون هذا الكلام صحيحا أيضا فيما يتعلق بالأرقام. غالبًا ما تكون قراءة البيانات الأولية صعبة ونادراً ما تعطينا نظرة واضحة للبيانات التي يمكن أن يوفرها التمثيل البياني الجيد للبيانات. الهدف من تحليل البيانات هو توفير رؤية عميقة للبيانات والتي يمكن توصيفها وشرحها للآخرين.
عادةً ما توفر تطبيقات جداول البيانات الحديثة آلية لعرض البيانات في جدول البيانات في أنواع

مختلفة من الرسوم البيانية ، مثل المخططات الخطية أو المخططات الدائرية. غالبًا ما يكون عدد أنواع الرسوم البيانية متنوع جداً ، ويسمح لك بتخصيص أفضل السبل لنقل الأفكار التي تريد التأكيد عليها. يظهر مثال للرسم البياني الذي تم إنشاؤه بواسطة Microsoft Excel في الشكل ٥. وكما هو الحال مع تحليل "ماذا لو" (What-if) إذا تغيرت البيانات سوف تتغير الرسوم البيانية تبعا لها.

	A	B	C
1	Date	Low (°F)	High (°F)
2	3/15/18	32	43
3	3/16/18	28	38
4	3/17/18	19	42
5	3/18/18	16	33
6	3/19/18	21	36
7	3/20/18	24	39
8	3/21/18	32	39
9	3/22/18	32	42
10	3/23/18	35	42
11	3/24/18	34	46
12	3/25/18	32	39
13	3/26/18	29	40
14	3/27/18	28	42
15	3/28/18	36	54
16	3/29/18	37	50
17	3/30/18	43	64
18	3/31/18	39	57



شكل ٥: مخطط بياني تم أنشاءه من بيانات جدول البيانات

مصدر الشكل كتاب Computer Science Illuminated by Nell B. Dale

٣ أنظمة أدارة قواعد البيانات (Database Management Systems)

تعتمد جميع حالات إدارة البيانات المعقدة تقريبًا على قاعدة بيانات أساسية وبنية داعمه تسمح للمستخدم (سواء أكان إنسانًا أم برنامجًا) بالتفاعل معها. يمكن ببساطة تعريف قاعدة البيانات على أنها مجموعة منظمة من البيانات. نظام إدارة قواعد البيانات (DBMS) هو مزيج من البرامج والبيانات المكونة من ثلاثة مكونات:

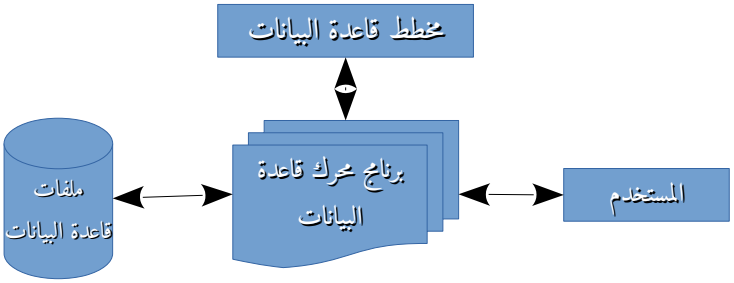
- قاعدة البيانات الفعلية - مجموعة من الملفات التي تحتوي على البيانات
- محرك قاعدة البيانات — برنامج يدعم الوصول إلى محتويات قاعدة البيانات وتعديلها
- مخطط قاعدة البيانات - مخطط للعلاقات التي تربط البيانات المخزنة في قاعدة البيانات

يتفاعل برنامج **محرك قاعدة البيانات** مع لغات قواعد البيانات المتخصصة التي تسمح للمستخدم بتحديد هيكل البيانات وإضافة وتعديل وحذف البيانات والاستعلام **(Query)** عن قاعدة البيانات لاسترداد بيانات مخزنة محددة.

يوفر **مخطط قاعدة البيانات (Database Schema)** العلاقات التي تربط البيانات الموجودة في قاعدة البيانات بغض النظر عن كيفية تخزينها فعليًا. بافتراض أن البنية الفعلية الأساسية لقاعدة

البيانات يتم بناءها بطريقة فعالة فإن مخطط قاعدة البيانات المنطقي هو وجهة النظر الأكثر أهمية من منظور مستخدم قاعدة البيانات لأنه يوضح كيفية ارتباط عناصر البيانات ببعضها البعض.

يصور الشكل ٦ العلاقات بين العناصر المختلفة لنظام إدارة قاعدة البيانات. يتفاعل المستخدم مع برنامج محرك قاعدة البيانات لتحديد و / أو تعديل مخطط قاعدة البيانات. ثم يتفاعل المستخدم مع برنامج المحرك للوصول وربما تعديل محتويات قاعدة البيانات المخزنة على القرص.



شكل ٦: عناصر أنظمة أدارة قواعد البيانات

مصدر الشكل كتاب Computer Science Illuminated by Nell B. Dale

٣.١ نموذج العلاقات (The Relational Model)

تم اقتراح العديد من نماذج إدارة قواعد البيانات الشائعة ، ولكن النموذج الذي سيطر على مدار سنوات عديدة هو نموذج العلاقات. في نظام قواعد البيانات العلائقية ، يتم تنظيم عناصر البيانات والعلاقات فيما بينها في جداول. الجدول عبارة عن مجموعة من السجلات (Records).

السجل عبارة عن مجموعة من الحقول (Fields) ذات الصلة. يحتوي كل حقل (Field) في جدول قاعدة البيانات على قيمة بيانات واحدة. يحتوي كل سجل (Records) في جدول على نفس الحقول (Fields). يُطلق أيضًا على السجل في جدول قاعدة البيانات اسم كائن (Object) أو كيان (Entity) قاعدة بيانات. تسمى حقول السجل أحيانًا بسمات (Attributes) كائن قاعدة البيانات.

على سبيل المثال انظر إلى جدول قاعدة البيانات الموضح في الشكل ٨ والذي يحتوي على معلومات حول الأفلام. كل صف يمثل سجل في الجدول. يتكون كل سجل في الجدول من نفس الحقول التي يتم فيها تخزين قيم معينة. أي أن كل سجل يمثل فيلم يحتوي على معرف الفيلم (MovieId) والعنوان (Title) والنوع (Genre) والتصنيف (Rating) والذي يحتوي على بيانات محددة لكل فيلم. يتم إعطاء اسم لجدول قاعدة البيانات ، مثل "Movie" في هذه الحالة.

عادةً ما يتم تحديد حقل واحد أو أكثر في الجدول كحقول مفتاحية (Key Fields).تكون سجلات الحقل المفتاحي فريدة (إي أن قيمها لا تتكرر ابداً) من بين جميع السجلات الأخرى في الجدول. أي

أن القيمة المخزنة في الحقول المفتاحية لكل سجل في الجدول يجب أن تكون فريدة لا تتكرر. في جدول الفيلم ، سيكون حقل MovieId هو الخيار المنطقي كحقل مفتاحي. بهذه الطريقة ، يمكن أن يحمل فيلمان نفس العنوان. من المؤكد أن حقلي "النوع" و "التصنيف" ليسا حقلين مفتاحيين مناسبين في هذه الحالة (لماذا؟).

يجب أن تكون كل قيمة في الحقل المفتاحي MovieId فريدة ولا تتكرر. تسمح معظم نظم إدارة قواعد البيانات (DBMS) بإنشاء مثل هذه الحقول تلقائيًا لضمان إدخالات فريدة. ومع ذلك ، لا يشترط أن تكون القيم المفتاحية متتالية ومتسلسلة. فمثلا نلاحظ ان قيم الحقل المفتاحي MovieId الثلاثة الأخيرة في الشكل ٧ على أرقام مختلفة وغير متسلسلة. طالما أنها قيم فريدة وغير متكررة اذن يمكن أن يعمل حقل MovieId كمفتاح.

يتم تقديم جدول الفيلم في الشكل ٧ بترتيب زيادة قيمة MovieId ، ولكن كان من الممكن عرضه بطرق أخرى ، مثل الترتيب الأبجدي حسب عنوان الفيلم. في هذه الحالة ، لا توجد علاقة متأصلة بين صفوف البيانات في الجدول. تقدم جداول قاعدة البيانات العلائقية عرضًا منطقيًا للبيانات وليس لها علاقة بالتنظيم المادي الأساسي (كيفية تخزين السجلات على قرص خزن البيانات). يصبح ترتيب السجلات (تصاعديا او تنازليا حسب الأحرف الأبجدية والأرقام) مهمًا فقط عندما نستعلم في قاعدة البيانات عن قيم معينة ، مثل جميع الأفلام التي تم تصنيفها على أنها PG. في هذه المرحلة ، قد نرغب في فرز نتائج الاستعلام حسب العنوان.

يتوافق بنية الجدول مع المخطط الذي يمثله. أي أن المخطط هو تعبير عن سمات السجلات في جدول. يمكننا التعبير عن مخطط هذا الجزء من قاعدة البيانات على النحو التالي:

Movie (MovieId:key, Title, Genre, Rating)

أحياناً يشير تمثيل المخطط (Schema) إلى نوع البيانات المخزنة في حقول فردية كان تكون رقمية أو نصية. قد يشير أيضًا إلى مجموعة معينة من القيم المناسبة لحقل معين. على سبيل المثال ، يمكن أن يشير المخطط في هذا المثال إلى أن حقل التصنيف يمكن أن يكون فقط G أو PG أو PG-13 أو R أو NC-17. يتكون المخطط الكلي لقاعدة البيانات من مجموعة مخططات تنتمي إلى مجموعة جدول في قاعدة البيانات.

لنفترض أننا أردنا إنشاء شركة لتأجير الأفلام. بالإضافة إلى قائمة الأفلام المعروضة للإيجار ، يجب

علينا إنشاء جدول قاعدة بيانات يحتوي على معلومات حول عملائنا. يمكن أن يمثل جدول العملاء في الشكل ٨ هذه المعلومات.

MovieId	Title	Genre	Rating
101	Sixth Sense, The	thriller horror	PG-13
102	Back to the Future	comedy adventure	PG
103	Monsters, Inc.	animation comedy	G
104	Field of Dreams	fantasy drama	PG
105	Alien	sci-fi horror	R
106	Unbreakable	thriller	PG-13
107	X-Men	action sci-fi	PG-13
5022	Elizabeth	drama period	R
5793	Independence Day	action sci-fi	PG-13
7442	Platoon	action drama war	R

شكل ٧: جدول قواعد بيانات الأفلام مكونة من سجلات وحقول

مصدر الشكل كتاب Computer Science Illuminated by Nell B. Dale

على غرار ما فعلناه مع جدول الأفلام ، يحتوي جدول العملاء على حقل CustomerId ليكون بمثابة الحقل المفتاحي. حقيقة أن بعض قيم CustomerId تتوافق مع بعض قيم MovieId ليست ذات صلة. يجب أن تكون القيم الأساسية فريدة فقط داخل الجدول.

في قاعدة البيانات الحقيقية ، سيكون من الأفضل تقسيم حقل الاسم في جدول العملاء إلى حقول الاسم الأول والاسم الأخير. كذلك من المحتمل أن نستخدم حقول منفصلة للعنوان الكامل مثل المدينة والولاية. بالنسبة لأمثلتنا فإننا نبقي الأمور بسيطة.

CustomerId	Name	Address	CreditCardNumber
101	Dennis Cook	123 Main Street	2736 2371 2344 0382
102	Doug Nickle	456 Second Ave	7362 7486 5957 3638
103	Randy Wolf	789 Elm Street	4253 4773 6252 4436
104	Amy Stevens	321 Yellow Brick Road	9876 5432 1234 5678
105	Robert Person	654 Lois Lane	1122 3344 5566 7788
106	David Coggin	987 Broadway	8473 9687 4847 3784
107	Susan Klaton	345 Easy Street	2435 4332 1567 3232

شكل ٨: جدول قاعدة بيانات العملاء الذي يحتوي على بيانات العملاء

مصدر الشكل كتاب Computer Science Illuminated by Nell B. Dale

يوضح جدول الأفلام وجدول العملاء كيف يمكن تنظيم البيانات كسجلات داخل جداول منفصلة. على الرغم من ذلك ، تكمن القوة الحقيقية لأنظمة إدارة قواعد البيانات العلائقية في القدرة على إنشاء جداول مربوطة مع جداول أخرى معًا بشكل منطقي.

ذكرنا أن السجلات تمثل كائنات قاعدة بيانات فردية وأن حقول السجل هي سمات لهذه السجلات. يمكننا إنشاء سجل لتمثيل العلاقة بين الكائنات وتضمين السمات الخاصة بالعلاقة في ذلك السجل. بهذه الطريقة ، يمكننا استخدام جدول لتمثيل مجموعة من العلاقات بين الكائنات. لتوضيح ذلك في مثال تأجير الأفلام ، نحتاج إلى أن نكون قادرين على تمثيل الموقف الذي يستأجر فيه عميل معين فيلمًا معينًا. ولأن "الإيجارات" هي علاقة بين العميل والفيلم فيمكننا تمثيلها كسجل. يعتبر تاريخ الإيجار وتاريخ الاستحقاق من سمات العلاقة التي يجب أن تكون في السجل. يحتوي جدول الإيجارات في الشكل ٩ على مجموعة من سجلات العلاقات التي تمثل الأفلام المستأجرة حاليًا.

يحتوي جدول الإيجارات على معلومات حول الكائنات في العلاقة (العملاء والأفلام) ، بالإضافة إلى سمات العلاقة. ومع ذلك ، فإنه لا يحتفظ بجميع البيانات المتعلقة بالعميل أو الفيلم. في قاعدة البيانات العلائقية ، نتجنب تكرار البيانات قدر الإمكان. على سبيل المثال ، ليست هناك حاجة لتخزين اسم العميل وعنوانه في جدول الإيجار - هذه البيانات مخزنة بالفعل في جدول العملاء. عندما نحتاج إلى هذه البيانات ، نستخدم معرف العميل المخزن في جدول الإيجارات للبحث عن بيانات العميل التفصيلية في جدول العملاء. كذلك عندما نحتاج إلى بيانات حول الفيلم الذي تم استئجاره فإننا نبحث عنه في جدول الافلام باستخدام MovieId. لاحظ أن القيمة 103 في الحقل CustomerId تظهر في سجلين في جدول الإيجارات. ويشير ظهورهما إلى أن نفس العميل قد استأجر فيلمين مختلفين.

CustomerId	MovieId	DateRented	DateDue
103	104	3-12-2018	3-13-2018
103	5022	3-12-2018	3-13-2018
105	107	3-12-2018	3-15-2018

شكل ٩: جدول قاعدة بيانات الإيجارات لخزن إيجارات الأفلام الحالية

مصدر الشكل كتاب Computer Science Illuminated by Nell B. Dale

يتم تعديل البيانات وإضافتها وحذفها من جداول قاعدة البيانات المختلفة حسب الحاجة. عند إضافة الأفلام أو إزالتها من المخزون المتاح نقوم بتحديث سجلات جدول الأفلام تبعا لهذه الاضافة او

الحذف. عندما يصبح الأشخاص عملاء جدد لمتجرنا ، نضيفهم إلى جدول العملاء. وبشكل مستمر نقوم بإضافة السجلات وإزالتها من جدول الإيجارات حين يقوم العملاء باستئجار مقاطع الفيديو ثم إعادتها.

٢.٣ لغة الاستعلامات المهيكلة (Structured Query Language)

أن لغة الاستعلام المهيكلة (SQL) هي لغة قاعدة بيانات شاملة لإدارة قواعد البيانات العلائقية. تتضمن عبارات تحدد مخطط قاعدة البيانات بالإضافة إلى عبارات إضافة وتعديل وحذف محتوى قاعدة البيانات. بالإضافة إلى ذلك ، كما يوحي اسمها ، توفر SQL القدرة على الاستعلام عن قاعدة البيانات لاسترداد بيانات محددة.

كان الإصدار الأصلي من SQL هو Sequal ، الذي طورته شركة IBM في أوائل السبعينيات. في عام ١٩٨٦ ، نشر المعهد الوطني الأمريكي للمعايير (ANSI) معيار SQL ، والذي يعمل كأساس للغات قواعد البيانات التجارية للنفاذ إلى قواعد البيانات العلائقية.

SQL ليست حساسة لحالة الأحرف ، لذلك يمكن أن تكون الكلمات الرئيسية وأسماء الجداول وأسماء السمات أحرف كبيرة أو صغيرة أو مختلطة. يتم استخدام المسافات كفواصل في عباراتها.

أولاً لنستعرض الاستعلامات البسيطة. عبارة التحديد "Select" هي الأداة الأساسية لهذا الغرض. تتضمن جملة select الأساسية جملة select وfrom وwhere:

```
select attribute-list from table-list where condition
```

حدد جملة التحديد السمات التي يتم إرجاعها. تحدد عبارة from الجداول المستخدمة في الاستعلام. تقوم جملة where بتقييد البيانات التي يتم إرجاعها. على سبيل المثال:

```
select Title from Movie where Rating = 'PG'
```

نتيجة هذا الاستعلام هي قائمة بجميع العناوين من جدول الأفلام التي لها تصنيف PG.

يمكن حذف شرط أين "Where" إذا لم تكن هناك ضرورة لقيود خاصة:

```
select Name, Address from Customer
```

يقوم هذا الاستعلام بإرجاع اسم وعنوان كافة العملاء في جدول العملاء.

يمكن استخدام علامة النجمة (*) في جملة التحديد "Select" للإشارة إلى أنه يجب إرجاع جميع السمات في السجلات المحددة:

select * from Movie where Genre like '%action%'

يعرض هذا الاستعلام جميع سمات السجلات من جدول الأفلام الذي تحتوي فيه سمة النوع (Genre) على الكلمة "action". يقوم المعامل like في لغة SQL بمطابقة الأنماط البسيطة على النصوص ويتطابق الرمز ٪ مع أي نص.

يمكن أن تحدد عبارات التحديد (Select) أيضًا كيفية فرز نتائج الاستعلام باستخدام الترتيب حسب الجملة:

select * from Movie where Rating = 'R' order by Title

يقوم هذا الاستعلام بإرجاع جميع سمات الأفلام المصنفة (Rated) من فئة R مرتبة حسب عنوان الفيلم. تدعم لغة SQL العديد من صيغ عبارات التحديد أكثر من التي تم توضيحها هنا.

تسمح عبارات SQL للإدراج والتحديث والحذف بتغيير البيانات الموجودة في الجدول. تضيف عبارة الإدراج (Insert) سجلاً جديدًا إلى الجدول. تحدد كل عبارة insert قيم سمات السجل الجديد. على سبيل المثال:

insert into Customer values (9876, 'John Smith, '602 Greenbriar Court', '2938 3212 3402 0299')

يُدرج هذا البيان سجلاً جديدًا في جدول العملاء بالسمات (القيم) المحددة.

ان عبارة التحديث (Update) تغير القيم في واحد أو أكثر من سجلات الجدول. على سبيل المثال:

update Movie set Genre = 'thriller drama' where title = 'Unbreakable'

يغير هذا البيان سمة النوع (Genre) لفيلم Unbreakable إلى دراما مثيرة (thriller drama).

تقوم عبارة الحذف (Delete) بإزالة جميع السجلات من الجدول المطابق للشرط المحدد. على سبيل المثال ، إذا أردنا إزالة جميع الأفلام ذات التصنيف R من جدول الأفلام ، فيمكننا استخدام عبارة الحذف التالية:

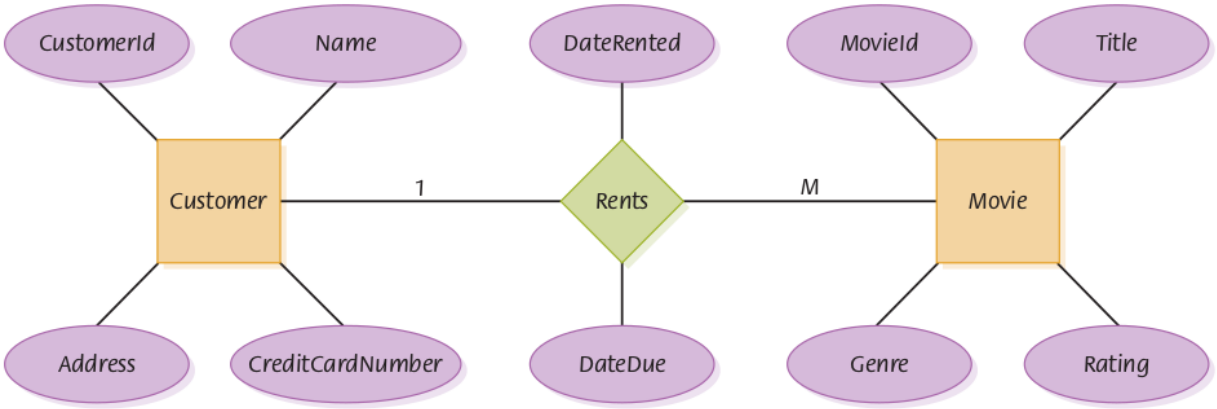
delete from Movie where Rating = 'R'

كما هو الحال مع عبارة select ، هناك العديد من الصيغ الأخرى في عبارات insert و update و delete.

٣-٣ تصميم قاعدة البيانات (Database Design)

يجب تصميم قاعدة البيانات بعناية من البداية إذا كانت تأمل في أداء دورها. يمكن أن يؤدي التخطيط السيئ في المراحل المبكرة إلى قاعدة بيانات لا تدعم العلاقات المطلوبة.

يُطلق على أحد الأساليب الشائعة لتصميم قواعد البيانات العلائقية نمذجة علاقة الكيان (Entity-Relationship-ER). من بين الأدوات المستخدمة في نمذجة ER هو **مخطط ER**. يحتوي مخطط ER أنواع السجلات المهمة والسمات والعلاقات في شكل رسومي. من مخطط ER يمكن لمدير قاعدة البيانات تحديد المخطط الضروري وإنشاء الجداول المناسبة لدعم قاعدة البيانات المحددة بواسطة المخطط. يعرض الشكل ١٠ مخطط ER موضحاً الجوانب المختلفة لمثال تأجير الأفلام. تُستخدم أشكال محددة في مخططات ER للتمييز بين الأجزاء المختلفة لقاعدة البيانات. يتم عرض أنواع السجلات (التي يمكن اعتبارها أيضًا فئات لكائنات قاعدة البيانات) في المستطيلات. تظهر الحقول (أو السمات) الخاصة بهذه السجلات في الأشكال البيضاوية المرفقة. تظهر العلاقات في الشكل المعيني. لا تعتبر مواضع العناصر المختلفة لمخطط ER مهمة بشكل خاص ، على الرغم من أن التفكير فيها سيجعل الرسم التخطيطي أسهل في القراءة. لاحظ أن علاقة مثل الإيجارات يمكن أن يكون لها سماتها الخاصة المرتبطة بها.



شكل ١٠: مخطط ER لقاعدة بيانات تأجير الأفلام

مصدر الشكل كتاب Computer Science Illuminated by Nell B. Dale

لاحظ أيضًا أن موصلات العلاقة مُصنَّفة ، أحد الجانبين به ١ والجانب الآخر بالحرف M. تُظهر هذه التعيينات قيود العلاقة الأساسية للعلاقة. يضع قيد العلاقة الأساسية قيودًا على عدد العلاقات التي قد توجد في وقت واحد. ثلاث علاقات أساسية عامة ممكنة:

- واحد لواحد (One-to-One)
- واحد لكثير (One-to-Many)

- الكثير للكثيرين (Many-to-Many)

ان العلاقة بين العميل والفيلم علاقة واحد لكثير. أي أنه يُسمح لعميل واحد باستئجار العديد من الأفلام ، ولكن لا يمكن استئجار الفيلم إلا لعميل واحد (في أي وقت). تساعد قيود العلاقة الأساسية مصمم قاعدة البيانات على نقل تفاصيل العلاقة.

٤ التجارة الإلكترونية (E-Commerce)

تتعامل تطبيقات الحاسوب في مجال التجارة الإلكترونية مع المعاملات المالية التي تتم عبر الويب. ويشمل جميع جوانب التسويق والمبيعات والشراء لكل من المنتجات والخدمات بالإضافة إلى الخدمات المالية الأخرى مثل المعاملات المصرفية. في هذه الأيام ، غالبًا ما يكون الويب هو الخيار الأول للأشخاص الذين يفكرون في الشراء.

عندما اصبح الويب في شائعا في عام ١٩٩٤ ، توقع الكثير من الناس أنه سيكون له تأثير كبير على الطريقة التي نؤدي بها أعمالنا. في الواقع لقد استغرق الأمر عدة سنوات لكي تحظى التجارة الإلكترونية بالثقة الكافية ، وتصبح عملية بما فيه الكفاية لترسيخ جذورها في ثقافتنا.

يعتبر الموقع Amazon.com أحد أقدم مواقع التجارة الإلكترونية على الويب لم يحقق ربحًا لسنوات عديدة. ولكن من خلال المثابرة على طفرات النمو (المؤلمة في بعض الأحيان) اصبح الخيار الأول للمتسوقين. يسمح موقع eBay (وهو موقع مزايدات شهير) لأي شخص ببيع منتجاته عبر الإنترنت ، حتى بدون وجود تصريح عمل رسمي؛ اليوم ، يقوم العديد من تجار التجزئة بإجراء معاملاتهم فقط من خلال موقع eBay. كانت الشركات مثل PayPal ، التي تجعل عملية الشراء عبر الإنترنت أسهل بكثير من خلال استخلاص التفاصيل المالية للمشتري ، هي أيضًا مفتاح نجاح التجارة الإلكترونية. في الواقع ، اشترت eBay PayPal في عام ٢٠٠٢ ؛ مثل العديد من المواقع على الإنترنت ، تستخدم eBay PayPal حصريًا كنظام دفع إلكتروني.

كان تطور التقنيات المستندة إلى الويب عاملاً رئيسياً في نجاح التجارة الإلكترونية والقوة الدافعة وراء بعضها. كانت القدرة على تطبيق عبر الإنترنت على توفير تفاعل معزز للمستخدم أمرًا بالغ الأهمية في هذا النمو ، وكذلك تطوير بروتوكولات آمنة وعوامل أخرى تسمح بالتحويل الآمن للأموال الإلكترونية.

تعد عربات التسوق الإلكترونية جزءًا أساسيًا من عملية التجارة الإلكترونية مما يسمح للمستخدمين

بالحفـاظ على مجموعة مسـتمرة من العناصر وشراء تلك العناصر في معاملة واحدة. تتعقب العديد من مواقع التجارة الإلكترونية مشتريات المستخدم وتقدم اقتراحات لعناصر أخرى قد يجدها المستخدم مثيرة للاهتمام. هذا جانب من جوانب التجارة الإلكترونية لا يمكن حصوله بسهولة بالمقارنة مع عمليات الشراء التقليدية من المتاجر.

يتمثل أحد الجوانب المهمة الأخرى لنجاح التجارة الإلكترونية في تطور فهم البائعين لكيفية تسوق المتسوقين. أن أفضل مواقع التجارة الإلكترونية لديها الآن وسائل تسمح للمستخدم بالبحث عن السلع ومقارنتها بطرق مختلفة. وبالتأكيد فان هذه الوسائل لا يمكن للمستخدم تجربتها عند زيارة متجر فعلي.

٥ البيانات الضخمة (Big Data)

يمكن أن تتعامل أنظمة المعلومات التي تمت مناقشتها في المواضيع السابقة مع الكثير من البيانات ولكن العصر الرقمي غيّر ذلك في نواحٍ عديدة. نقوم بإنشاء البيانات مع كل إجراء نتخذه تقريبًا. عندما نشتري شيئًا ما عبر الإنترنت او عندما ننشره على وسائل التواصل الاجتماعي او عندما نحمل هواتفنا بإمكانيات تتبع نظام تحديد المواقع العالمي (GPS) فسوف نترك أثراً رقميًا. يعتقد الخبراء أن في كل يومين ينتج البشر بشكل جماعي نفس القدر من البيانات كما فعلنا من بداية الوقت حتى عام ٢٠٠٠. أدى هذا الموقف إلى ظهور مصطلح **البيانات الضخمة (Big Data)** والذي يشير إلى مجموعات البيانات الكبيرة والمعقدة لدرجة أن أنظمة المعلومات التي ناقشناها حتى الآن مثل جداول البيانات وقواعد البيانات غير كافية للتعامل معها.

من فوائد دراسة وتحليل البيانات الضخمة هي الحصول على رؤى عميقة عن الكثير من الأمور. كلما زادت معرفتك بشيء ما كان بإمكانك إدارته وتوجيهه بشكل أفضل كما تريد. تسمح لنا تقنيات البيانات الضخمة بمقارنة المزيد من البيانات وكشف العلاقات التي كانت مخفية في السابق.

فوائد البيانات الضخمة عديدة للأعمال التجارية. يمكن أن تكشف الرؤى الإضافية عما يريد العملاء شرائه ومتى ، ويمكن أن تساعد الشركات على العمل بكفاءة أكبر. في المستقبل ستتمتع الشركات التي تتبنى وجهة نظر تعتمد على البيانات وترى البيانات كأصل إستراتيجي بميزة مميزة على تلك التي لا تفعل ذلك.

البيانات الضخمة لها تأثير كبير يتجاوز نماذج الأعمال التقليدية. تستخدم مؤسسات الرعاية الصحية حلول البيانات الضخمة لتحليل أعداد هائلة من السجلات الطبية والصور ، مما يساعد في تشخيص

الأمراض وتطوير أدوية جديدة. يتم استخدام كميات هائلة من بيانات أجهزة الاستشعار للتنبؤ بالكوارث الطبيعية والاستجابة لها. تستخدم قوات الشرطة كلاً من مجموعات البيانات العامة والخاصة للتنبؤ بالجرائم ومنعها.

لكن تلك البصيرة الجديدة لها ثمن. هناك العديد من التحديات عندما يتعلق الأمر بالبيانات الضخمة:

- غالبًا ما تحتوي مجموعات البيانات الكبيرة على أنواع مختلفة من البيانات - ليس فقط البيانات الرقمية الأساسية والنصوص والصور ، ولكن أيضًا عناصر الوسائط المتعددة مثل الصوت والفيديو والرسوم المتحركة ولذا يتطلب تخزين هذه الكميات الكبيرة من البيانات كميات هائلة من الذاكرة الثانوية.
- يتطلب اختيار وتنظيم تلك البيانات بحيث يمكن تحليلها وتحويلها إلى معلومات ذات مغزى قوة معالجة كبيرة.
- تتطلب حماية البيانات الضخمة التي غالبًا ما تحتوي على معلومات شخصية حساسة تخطيطًا تقنيًا ولوجستيًا دقيقًا.

يجب أن يكون نظام المعلومات الذي تم إعداده للتعامل مع البيانات الضخمة قادرًا على التوسع بسرعة وبشكل مناسب ؛ أي ان مع وصول البيانات الجديدة يجب أن تكون التقنيات المستخدمة في نظام المعلومات قادرة على التعامل مع الحمل. إذا لم يتمكن نظام المعلومات من التوسع بشكل كافٍ فإنه سيحد من كيفية استخدام البيانات إن لم يتم إيقافها.

لذلك تتطلب القدرة على التعامل مع البيانات الضخمة تخطيطًا دقيقًا وحلولًا حسابية محددة. غالبًا ما تتضمن عملية آلية تنشئ عمليات محاكاة بناءً على البيانات وإعادة تشغيل تلك المحاكاة حتى تظهر الأنماط.

من المحتمل أن تلعب جداول البيانات وقواعد البيانات دائمًا دورًا مهمًا في الحوسبة. لا تتطلب كل المشكلات حلول للبيانات الضخمة. ومع ذلك من الأهمية بمكان فهم الدور المتزايد باستمرار الذي تلعبه البيانات الضخمة في مستقبل الحوسبة.