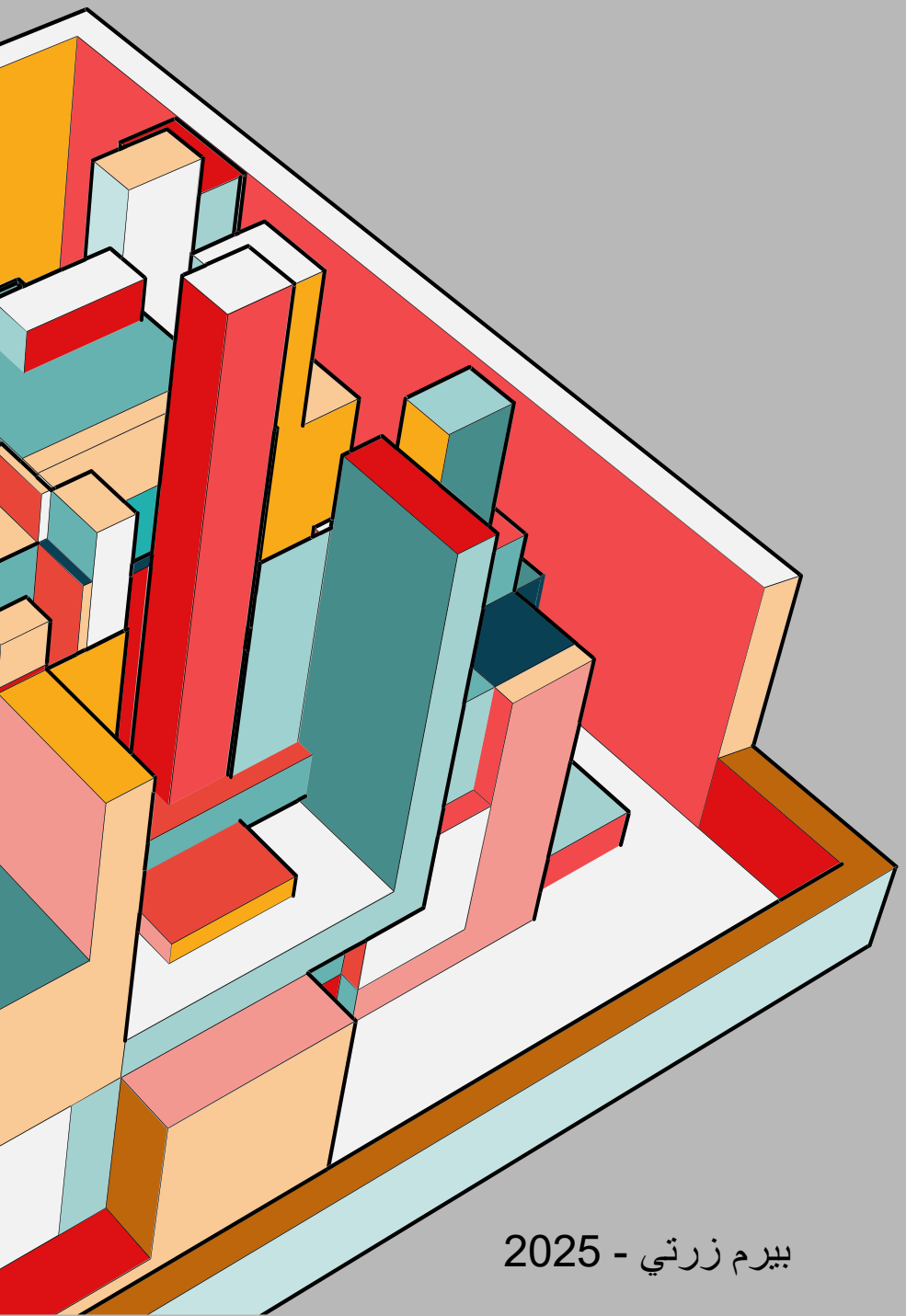


تصميم وهيكلة البرمجيات
ITSE411

المحاضرة السابعة

Process and Data Modeling

بيرم زرتي - 2025

KEY DEFINITIONS

- A *process model* is a formal way of representing how a business operates
- *Data flow diagramming* shows business processes and the data that flows between them
- *Logical process* models describe processes without suggesting how they are conducted
- Physical models include information about how the processes are implemented

نموذج العملية Process Model هو أسلوب رسمي لتمثيل كيفية عمل الأعمال أو الأنشطة داخل المؤسسة.

مخطط تدفق البيانات Data Flow Diagramming

يُستخدم لتوضيح العمليات التجارية والبيانات التي تتدفق بينها.

يركز على **ما يحدث** من تدفق معلومات بين الأنشطة المختلفة.

النماذج المنطقية Logical Process Models

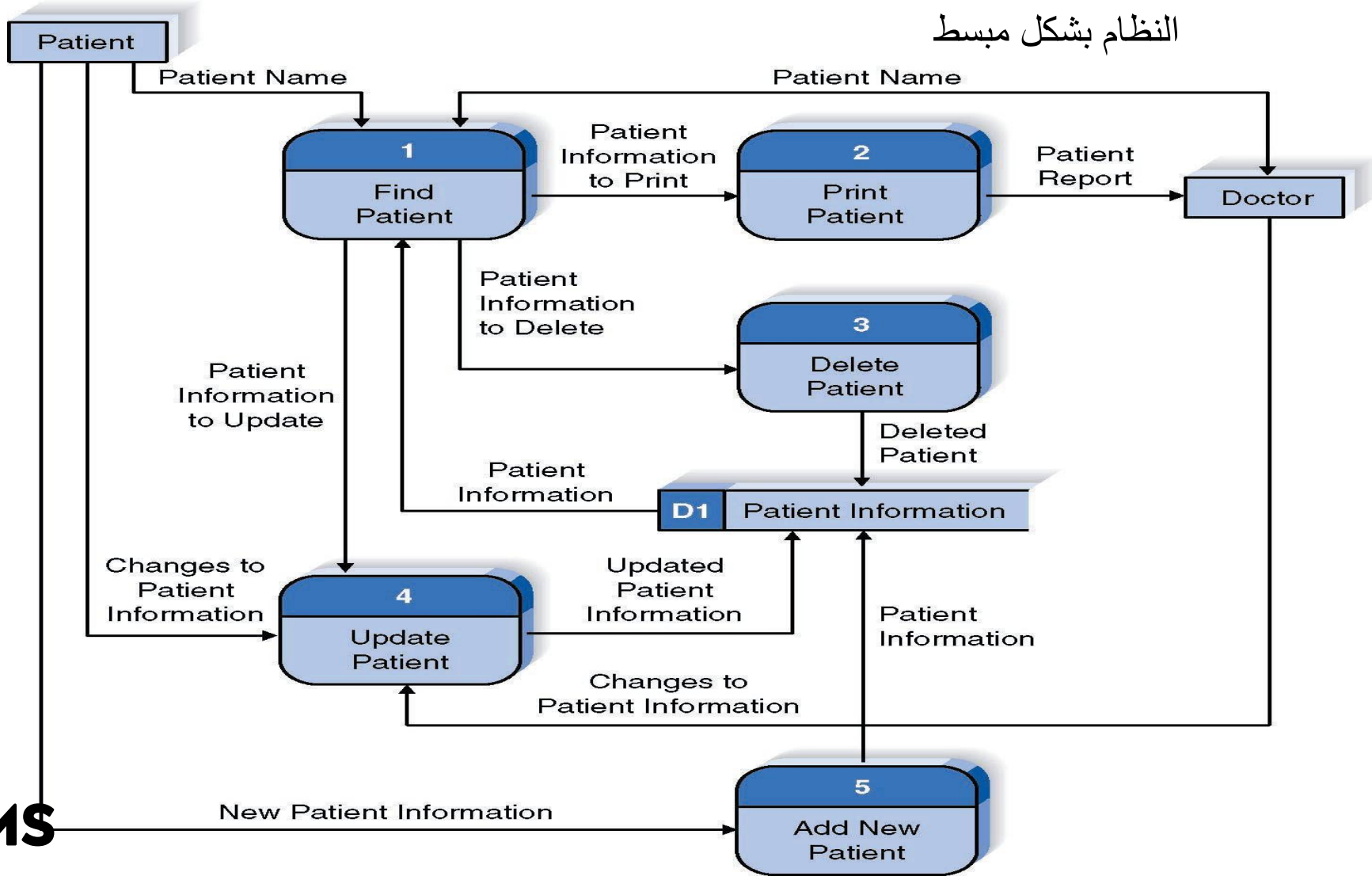
تصف العمليات دون الإشارة إلى كيفية تنفيذها فعليًا. تهدف إلى تقديم رؤية مجردة Abstract للأنشطة، بعيدًا عن التفاصيل التقنية أو التنفيذية.

النماذج الفيزيائية Physical Process Models

تتضمن معلومات حول **كيفية تنفيذ العمليات** بالفعل. تُظهر التفاصيل التقنية مثل الأنظمة، الأدوات، أو الأشخاص المسؤولين عن التنفيذ.

- ✓ **النموذج المنطقي** يجيب على سؤال: *ما هي العملية؟*
- ✓ **النموذج الفيزيائي** يجيب على سؤال: *كيف تُنفذ العملية؟*
- ✓ أما **مخطط تدفق البيانات** فهو أداة لعرض العمليات والبيانات المتبادلة بينها بشكل بصري.

- **مخطط تدفق البيانات** Data Flow Diagram – DFD **هو رسم بياني يُستخدم لتمثيل تدفق البيانات داخل نظام أو عملية بشكل بصري، مما يساعد على فهم كيفية دخول البيانات ومعالجتها وتخزينها وخروجها من النظام.** DFD تمثيل مرئي لتدفق البيانات بين العمليات المختلفة في النظام أو المؤسسة. يُستخدم بكثرة في **تحليل الأنظمة** و**الهندسة البرمجية** و**إدارة الأعمال** لأنه يوضح كيفية عمل النظام بشكل مبسط



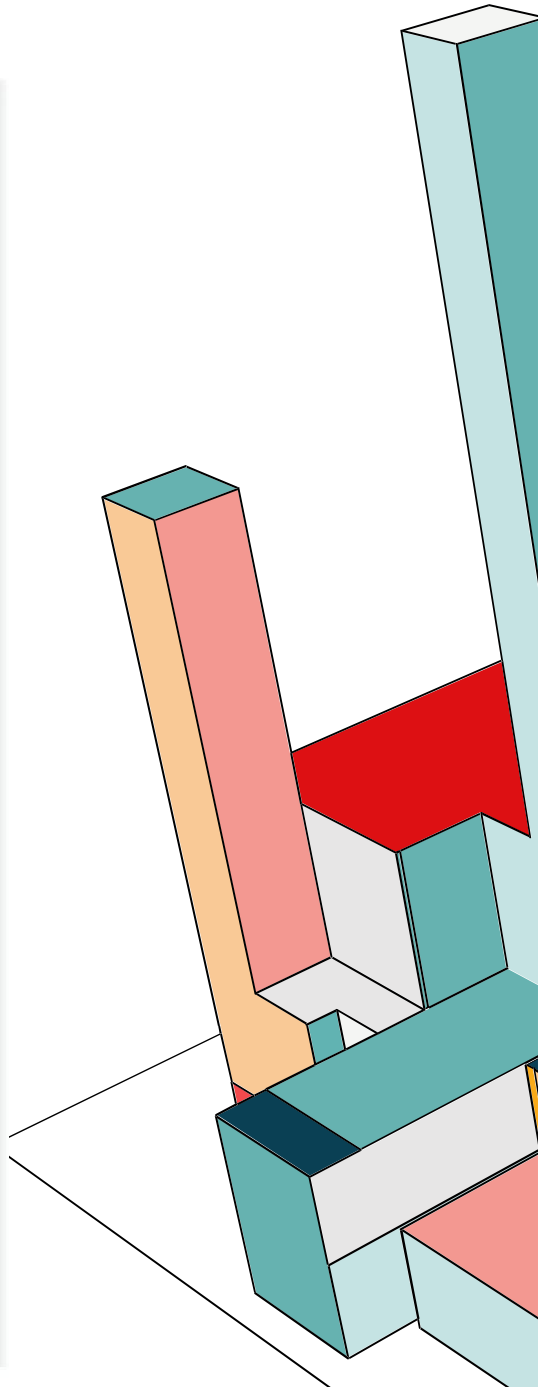
READING A DFD

DATA FLOW DIAGRAMS

DFD ELEMENTS

DATA FLOW DIAGRAMS

Data Flow Diagram Element	Typical Computer-Aided Software Engineering Fields	Gane and Sarson Symbol	DeMarco and Yourdan Symbol
Every *process* has A number A name (verb phase) A description One or more output data flows Usually one or more input data flows	Label (name) Type (process) Description (what is it) Process number Process description (Structured English) Notes	Name	Name
Every *data flow* has A name (a noun) A description One or more connections to a process	Label (name) Type (flow) Description Alias (another name) Composition (description of data elements) Notes	Name →	Name →
Every *data store* has A number A name (a noun) A description One or more input data flows Usually one or more output data flows	Label (name) Type (store) Description Alias (another name) Composition (description of data elements) Notes	D1 | Name	D1 Name
Every *external entity* has A name (a noun) A description	Label (name) Type (entity) Description Alias (another name) Entity description Notes	Name	Name



KEY DEFINITION

- *Decomposition* is the process of modeling the system and its components in increasing levels of detail.
- *Balancing* involves insuring that information presented at one level of a DFD is accurately represented in the next level DFD.

التفكيك Decomposition

- هو عملية **نمذجة النظام ومكوناته بمستويات متزايدة من التفاصيل**. يبدأ عادةً بمخطط سياقي Context Diagram يوضح النظام كعملية واحدة، ثم يتم تفصيل هذه العملية إلى عمليات فرعية في مستويات أدنى .... Level 1, Level 2 الهدف هو جعل النظام أكثر وضوحًا عبر تقسيمه إلى أجزاء أصغر وأسهل للفهم والتحليل.

الموازنة Balancing

- تعني التأكد من أن **المعلومات المعروضة في مستوى معين من مخطط تدفق البيانات** DFD يتم تمثيلها بدقة في المستوى التالي. أي أن المدخلات والمخرجات التي تظهر في المستوى الأعلى يجب أن تتطابق مع المدخلات والمخرجات في المستويات الأدنى. هذا يضمن الاتساق بين مستويات المخطط ويمنع فقدان أو إضافة بيانات غير مبررة أثناء التفصيل.

- ✓ **التفكيك** = زيادة التفاصيل تدريجيًا عبر مستويات متعددة.
- ✓ **الموازنة** = ضمان الاتساق والدقة بين هذه المستويات.
- ✓ كلا المفهومين ضروريان لبناء مخطط تدفق بيانات صحيح وموثوق يعكس النظام بشكل واقعي.

CONTEXT DIAGRAM

- Shows the context into which the business process fits
- Shows the overall business process as just *one* process
- Shows all the outside entities that receive information from or contribute information to the system

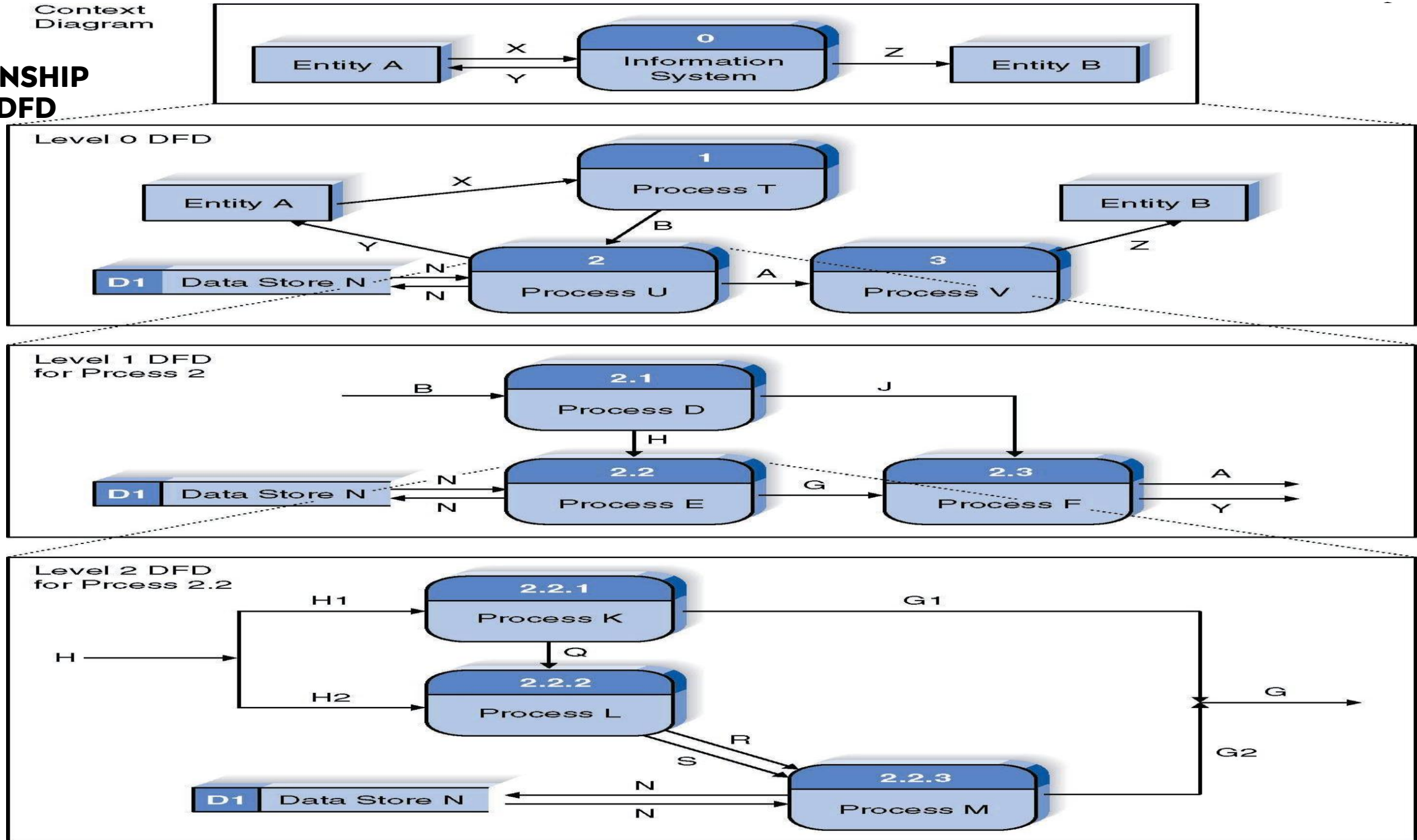
مخطط السياق هو صورة عامة تُظهر النظام ككتلة واحدة وعلاقته بالجهات الخارجية، قبل الدخول في تفاصيل العمليات الداخلية.

- يُظهر **السياق العام** الذي تنتمي إليه العملية أو النظام.
- يُمثل العملية التجارية أو النظام كعملية واحدة فقط One Process .
- يُوضح جميع **الكيانات الخارجية** External Entities التي تستقبل معلومات من النظام أو تُرسل معلومات إليه.
- أبسط مستوى من مخططات تدفق البيانات Level 0 .
- لا يحتوي على تفاصيل داخلية للعمليات، بل يركز على **العلاقات بين النظام والبيئة الخارجية**.
- يُستخدم كبداية لفهم النظام بشكل عام قبل الدخول في تفاصيل العمليات الفرعية.

مثلا في نظام تسوق إلكتروني:

- النظام يُعرض كعملية واحدة: **"إدارة الطلبات"**.
- الكيانات الخارجية: **العميل** (يرسل طلب شراء ويستقبل تأكيد)، **خدمة الدفع** (تتحقق من البطاقة)، **خدمة البريد الإلكتروني** (ترسل رسالة تأكيد).
- الأسهم تُظهر تدفق البيانات بين هذه الكيانات والنظام.

RELATIONSHIP AMONG DFD LEVELS



LEVEL 0 DIAGRAM

- Shows all the processes that comprise the overall system
- Shows how information moves from and to each process
- Adds data stores

مخطط المستوى صفر Level 0 Data Flow Diagram

- يُظهر **جميع العمليات الرئيسية** التي يتكون منها النظام ككل.
- يوضح **كيف تتحرك البيانات** بين العمليات المختلفة، وبين النظام والكيانات الخارجية.
- يضيف **مخازن البيانات** Data Stores التي تُستخدم لتخزين المعلومات بشكل دائم أو مؤقت.
- **الفرق عن مخطط السياق** Context Diagram
- **مخطط السياق** Context Diagram يُظهر النظام كعملية واحدة فقط مع المدخلات والمخرجات والكيانات الخارجية.
- **مخطط المستوى صفر** Level 0 DFD يُفكك العملية الرئيسية إلى عمليات فرعية متعددة، ويُظهر تدفق البيانات بينها، بالإضافة إلى مخازن البيانات.
- Level 0 DFDهو أول مستوى تفصيلي بعد مخطط السياق.
- يُظهر العمليات الأساسية، تدفق البيانات، ومخازن البيانات.
- يُستخدم لفهم النظام بشكل أوضح قبل الانتقال إلى مستويات أدق Level 1, Level 2…

LEVEL 1 DIAGRAMS

- Shows all the processes that comprise a single process on the level 0 diagram
- Shows how information moves from and to each of these processes
- Shows in more detail the content of higher level process
- Level 1 diagrams may not be needed for all level 0 processes

مخططات المستوى الأول Level 1 Data Flow Diagrams

- هي تفصيل لعملية واحدة من العمليات التي ظهرت في **مخطط المستوى صفر** Level 0 DFD تُظهر جميع العمليات الفرعية التي تُكوّن تلك العملية الكبرى. توضّح بشكل أدق **كيف تتحرك البيانات من وإلى كل عملية فرعية**. تُعطي تفاصيل أكثر عن محتوى العملية ذات المستوى الأعلى.
- كل عملية رئيسية في مخطط المستوى صفر يمكن تفكيكها إلى مخطط مستوى أول خاص بها. ليس من الضروري أن يتم تفصيل جميع العمليات؛ فقط العمليات المعقدة أو المهمة قد تحتاج إلى مخطط مستوى أول. تساعد في فهم تفاصيل النظام بشكل تدريجي دون فقدان الاتساق مع المستويات الأعلى مبدأ **الموازنة** Balancing
- Level 0 DFD يُظهر العمليات الرئيسية للنظام + تدفق البيانات + مخازن البيانات.
- Level 1 DFD يُفكك عملية واحدة من المستوى صفر إلى عمليات فرعية ويُظهر تفاصيلها.
- يُستخدم عند الحاجة لمزيد من التفاصيل، وليس إلزاميًا لكل العمليات.

LEVEL 2 DIAGRAMS

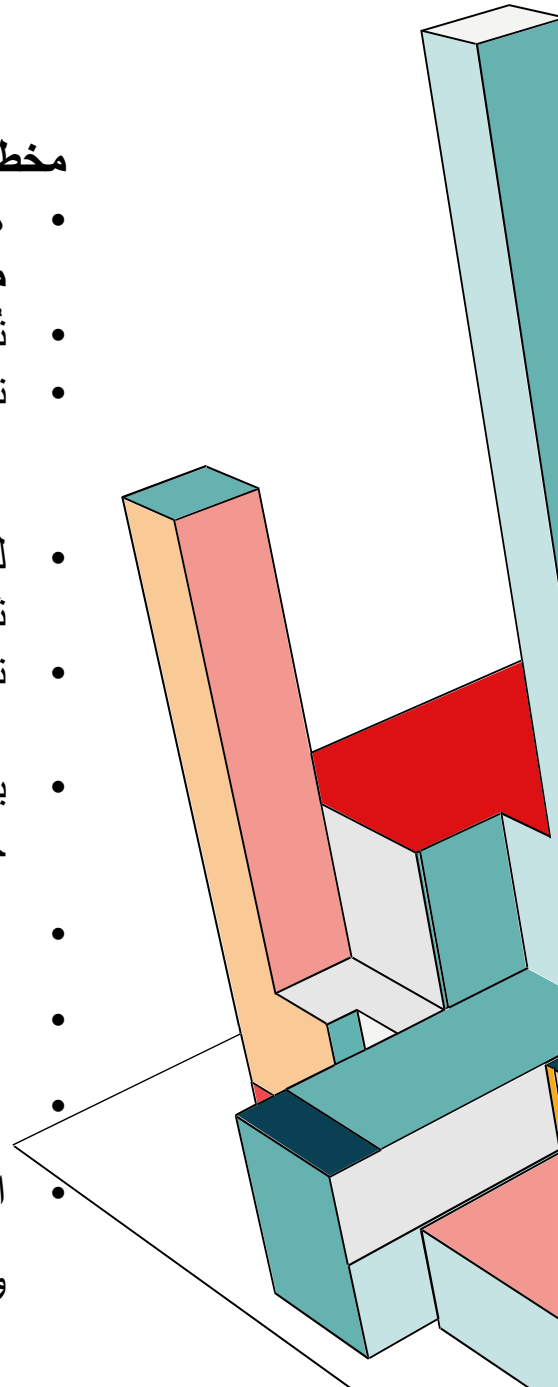
- Shows all processes that comprise a single process on the level 1 diagram
- Shows how information moves from and to each of these processes
- Level 2 diagrams may not be needed for all level 1 processes
- Correctly numbering each process helps the user understand where the process fits into the overall system

مخططات المستوى الثاني Level 2 DFDs

- هي تفصيل إضافي لعملية واحدة من العمليات التي ظهرت في **مخطط المستوى الأول** Level 1 DFD
- تُظهر جميع العمليات الفرعية التي تُكوّن تلك العملية المحددة.
- توضّح بشكل أدق **كيف تتحرك البيانات من وإلى كل عملية فرعية**.

- ليست كل العمليات في المستوى الأول تحتاج إلى تفصيل بمستوى ثاني؛ فقط العمليات المعقدة أو التي تتطلب فهمًا أدق.
- تساعد في بناء صورة أكثر وضوحًا للنظام عبر **التفكيك التدريجي** Decomposition
- يجب أن يتم **ترقيم العمليات بشكل صحيح** (مثل 3.2.1، 3.2.2 ...) حتى يفهم المستخدم مكان العملية ضمن النظام الكلي.

- Level 0 DFD العمليات الرئيسية للنظام.
- Level 1 DFD تفصيل عملية رئيسية إلى عمليات فرعية.
- Level 2 DFD تفصيل عملية فرعية من المستوى الأول إلى تفاصيل أدق.
- الترقيم الصحيح للعمليات (مثل 3.2.1، 3.2.2) يوضح موقعها في النظام ويسهل تتبعها.



INTEGRATING SCENARIO DESCRIPTIONS

- DFDs generally integrate scenario descriptions
- Names of use cases become processes
- Names of inputs and outputs become data flows
- Combining “small” data inputs and outputs into a single flow

- مخططات تدفق البيانات عادةً ما تُدمج مع **أوصاف السيناريوهات** Scenario Descriptions لتوضيح كيفية عمل النظام بشكل عملي. هذا الدمج يساعد على الانتقال من **النصوص الوصفية** مثل حالات الاستخدام Use Casesإلى **تمثيل رسومي** يوضح تدفق البيانات والعمليات.

أسماء حالات الاستخدام Use Cases

تتحول إلى **عمليات** Processes داخل مخطط تدفق البيانات.

مثال: حالة استخدام "تقديم طلب شراء" تصبح عملية رئيسية في DFD

أسماء المدخلات والمخرجات Inputs & Outputs

تتحول إلى **تدفقات بيانات** Data Flows

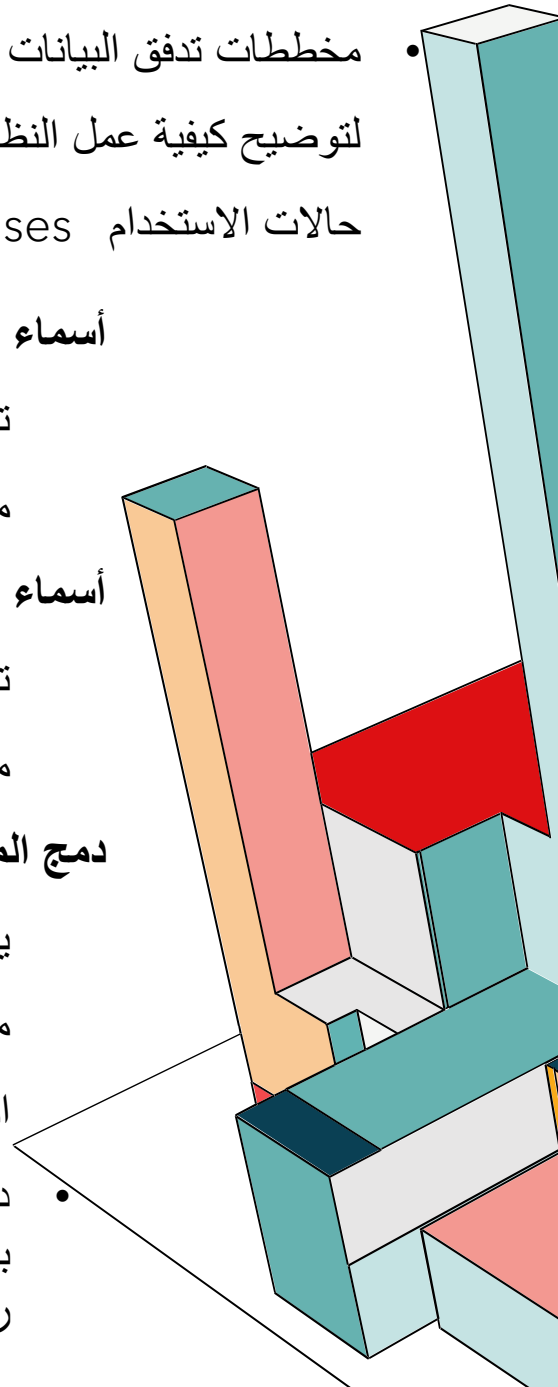
مثال: "طلب شراء" كمدخل، و"تأكيد الطلب" كمخرج.

دمج المدخلات والمخرجات الصغيرة Small Inputs/Outputs

يتم تجميعها في **تدفق واحد** لتبسيط المخطط وتجنب التعقيد.

مثال: بدلاً من رسم تدفق منفصل لكل معلومة صغيرة (اسم العميل، رقم البطاقة، العنوان)، يمكن دمجها في تدفق واحد باسم "بيانات العميل".

- دمج أوصاف السيناريوهات مع DFDيجعل المخطط أكثر وضوحًا وارتباطًا بالواقع العملي. يساعد على تحويل النصوص Use Cases إلى مخططات رسومية تُظهر العمليات، البيانات، والمخرجات بشكل متكامل.



STEPS IN BUILDING DFDS

- Build the context diagram
- Create DFD fragments for each scenario
- Organize DFD fragments into level 0
- Decompose level 0 DFDs as needed
- Validate DFDs with user

بناء مخطط السياق Context Diagram يُظهر النظام كعملية واحدة فقط. يوضح المدخلات والمخرجات الرئيسية والكيانات الخارجية التي تتفاعل مع النظام.

إنشاء أجزاء DFDلكل سيناريو DFD Fragments لكل سيناريو أو حالة استخدام Use Case يتم إنشاء جزء Fragment يوضح العمليات والبيانات الخاصة به. هذه الأجزاء تُعتبر اللبنات الأساسية لبناء المخطط الكامل.

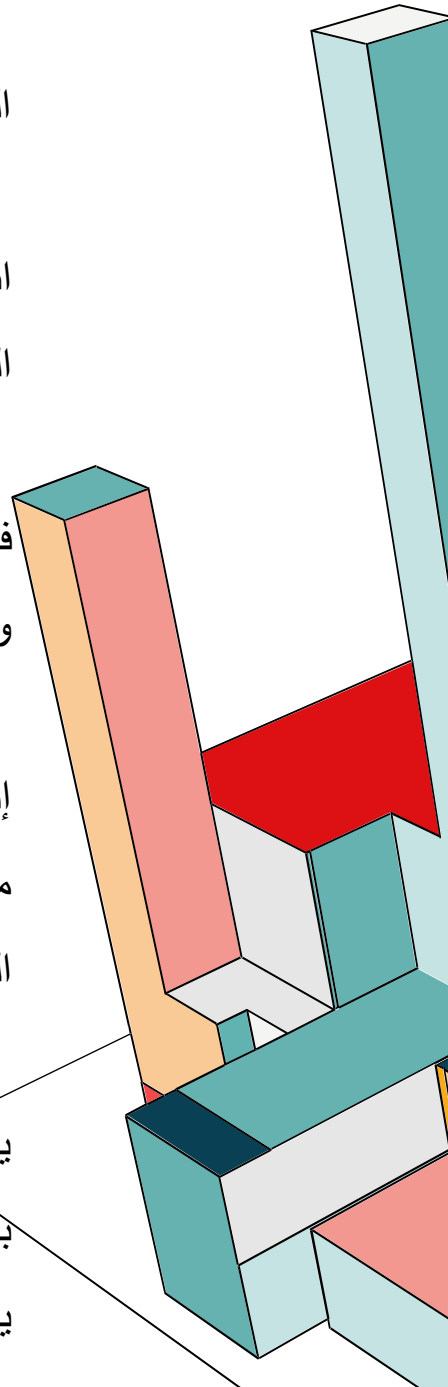
تنظيم الأجزاء في مخطط المستوى صفر Level 0 DFD يتم دمج الأجزاء المختلفة في مخطط شامل يمثل النظام ككل. يُظهر جميع العمليات الرئيسية، تدفق البيانات بينها، ومخازن البيانات.

تفكيك مخطط المستوى صفر عند الحاجة Decompose Level 0 DFDs

إذا كانت بعض العمليات معقدة، يتم تفصيلها في مخططات مستوى أول Level 1 أو مستوى ثاني Level 2 هذا التفكيك يُظهر تفاصيل أدق مع الحفاظ على الاتساق بين المستويات Balancing

التحقق من صحة المخططات مع المستخدم Validate DFDs with User

يتم عرض المخططات على المستخدم أو صاحب المصلحة للتأكد من أنها تعكس النظام بشكل صحيح. يساعد ذلك على اكتشاف الأخطاء أو الفجوات مبكرًا وضمان أن النموذج يعكس الواقع العملي.



DFD FRAGMENT TIPS

- All process names must be verb phrases
- Maintain organization's viewpoint in naming processes
- Layouts often place
  - processes in the center
  - inputs from the left
  - outputs to the right
  - stores beneath the processes

يجب أن تكون **أسماء العمليات عبارة عن عبارات فعلية.(Verb Phrases)**

مثال: "معالجة الطلب(Process Order) "، "التحقق من الدفع.(Verify Payment) "

هذا يعكس أن العملية تقوم **بنشاط أو فعل محدد** داخل النظام.

يجب الحفاظ على وجهة نظر المؤسسة عند تسمية العمليات .أي أن الأسماء يجب أن تُعبر عما يراه المستخدم أو المؤسسة، وليس عن تفاصيل تقنية داخلية.

مثال: "تسجيل عميل جديد" بدلاً من "إدخال بيانات في قاعدة البيانات."

تنظيم المخطط(Layout)

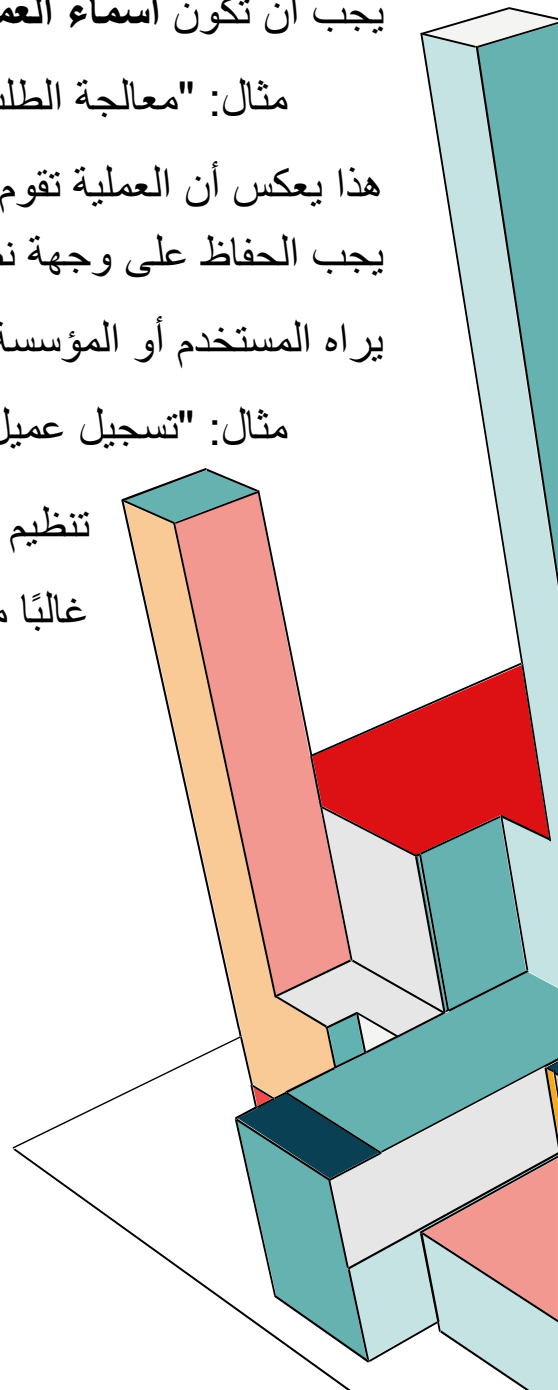
غالبًا ما يتم ترتيب العناصر في المخطط كما يلي:

العمليات :(Processes) في الوسط.

المدخلات :(Inputs) من اليسار.

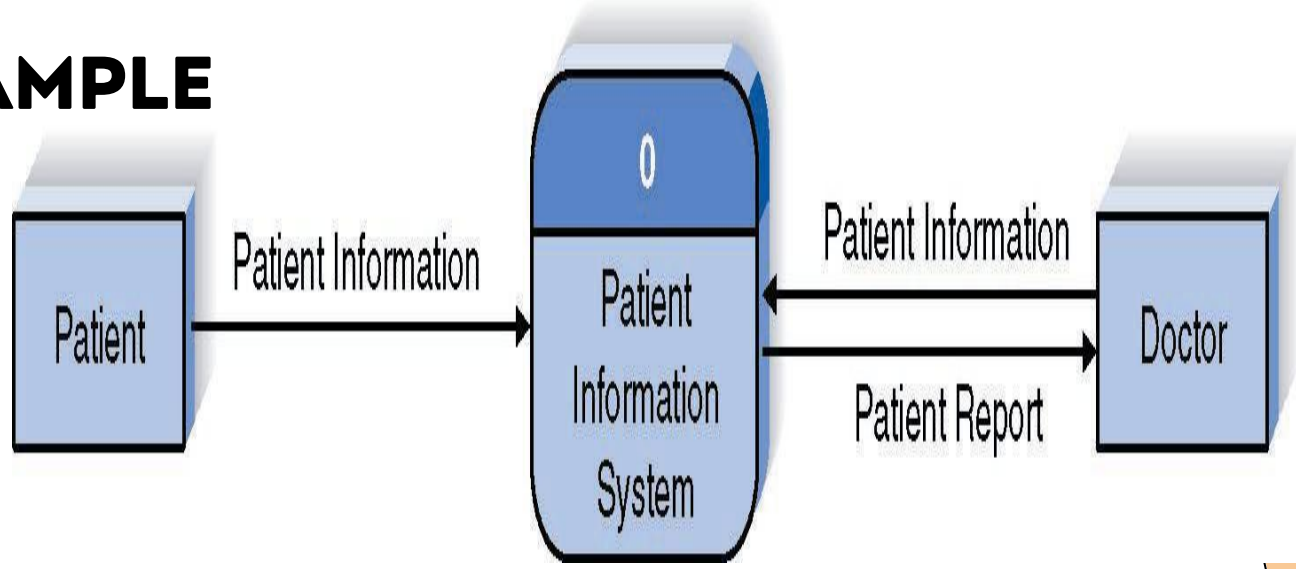
المخرجات :(Outputs) إلى اليمين.

مخازن البيانات :(Data Stores) أسفل العمليات.

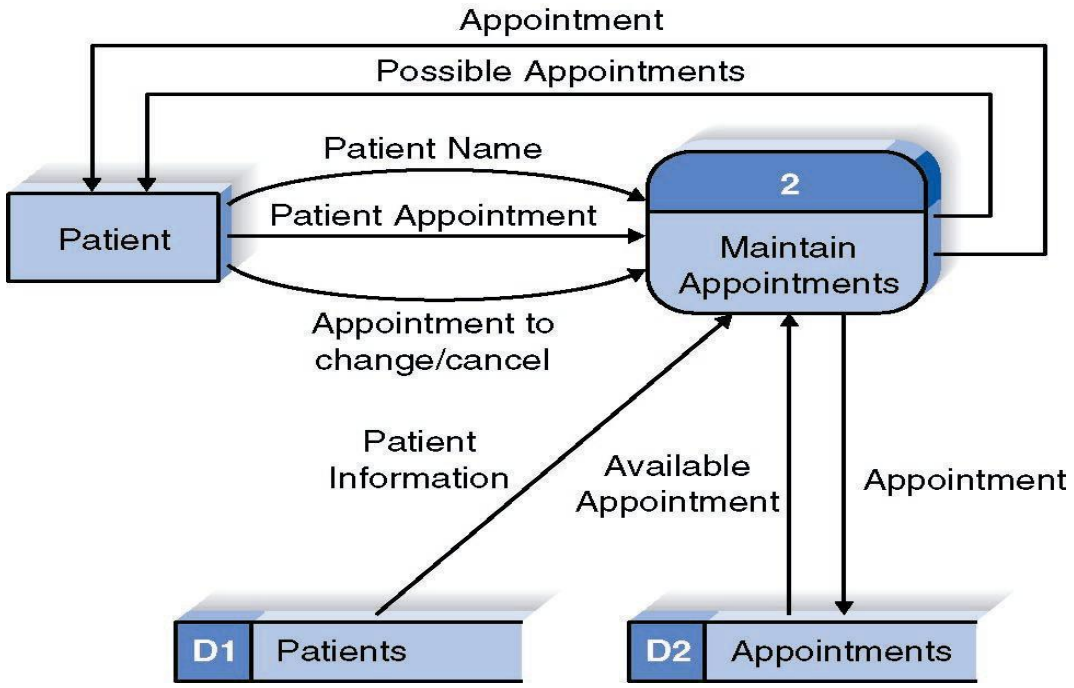


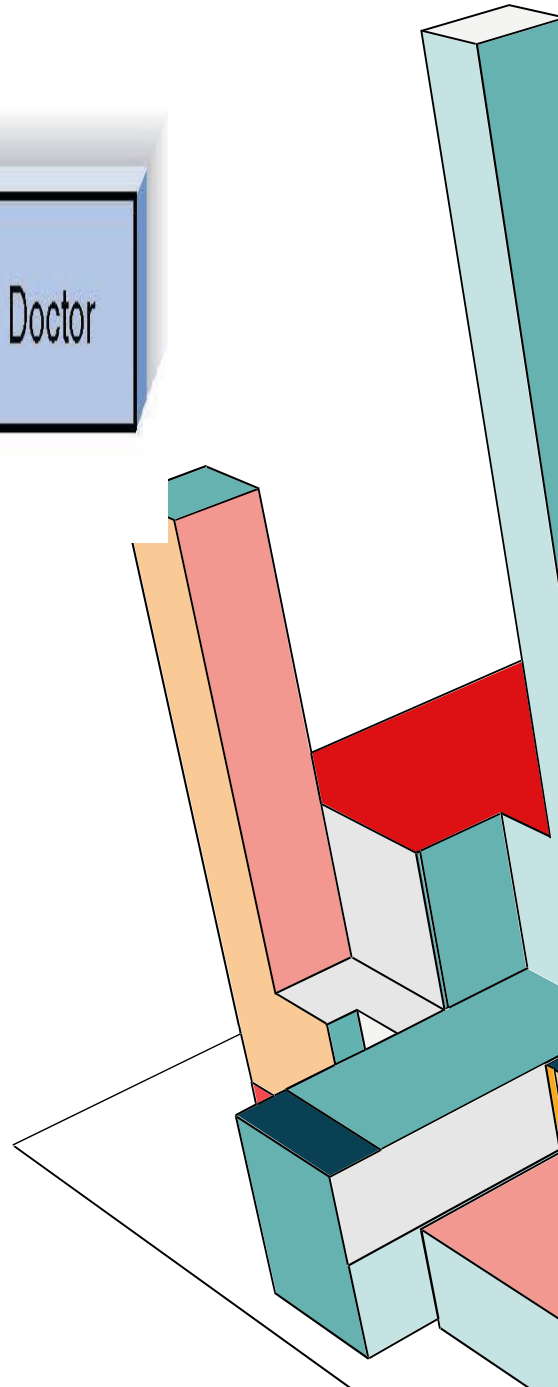
- تبدأ العملية من **الصورة العامة** Context Diagram
- ثم تُبنى تدريجيًا عبر **الأجزاء** Fragments و**المستوى صفر** Level 0
- يتم التفصيل عند الحاجة عبر **المستويات الأدنى** Level 1, Level 2
- وأخيرًا، يتم **التحقق مع المستخدم** لضمان دقة النموذج.

A DFD FRAGMENT EXAMPLE



A SECOND DFD FRAGMENT EXAMPLE





LEVEL 0 TIPS

- Generally move from top to bottom, left to right
- Minimize crossed lines
- Iterate as needed
  - *The DFD is often drawn many times before it is finished, even with very experienced systems analysts*

نصائح تصميم مخطط المستوى صفر Level 0 DFD

الترتيب العام

- عادةً يتم رسم المخطط **من الأعلى إلى الأسفل** و**من اليسار إلى اليمين**.
- هذا الترتيب يجعل قراءة المخطط أسهل وأكثر وضوحًا للمستخدمين والمحللين.

تقليل التعقيد

- حاول **تقليل الخطوط المتقاطعة** Crossed Lines قدر الإمكان.
- يمكن تحقيق ذلك عبر إعادة ترتيب العمليات أو الكيانات الخارجية لتوضيح التدفق بشكل أنظف.

التكرار والتحسين

- عملية رسم مخطط المستوى صفر غالبًا ما تحتاج إلى **عدة محاولات** Iterations قبل الوصول إلى الشكل النهائي.
- حتى المحللين ذوي الخبرة يقومون بإعادة رسم المخطط مرات عديدة لضمان الوضوح والدقة.

TIPS FOR LEVEL 1 AND BELOW

- Sources for inputs and outputs listed at higher level
- List source and destination of data flows to processes and stores within each DFD
- Depth of DFD depends on overall system complexity
  - Two processes generally don't need lower level
  - More than seven processes become overly complex and difficult to read

Level 1 and Below DFDs نصائح لمخططات المستوى الأول وما دونه

المدخلات والمخرجات

- يجب أن تكون **مصادر المدخلات والمخرجات** واضحة ومُدرجة في المستوى الأعلى Level 0
- عند الانتقال إلى مستوى أدنى، يتم تحديد **المصدر والوجهة** لكل تدفق بيانات Data Flow سواء كان إلى عملية أو إلى مخزن بيانات.

العمق Depth عمق مخطط تدفق البيانات يعتمد على **تعقيد النظام ككل**. ليس كل عملية تحتاج إلى تفصيل إضافي؛ فقط العمليات المعقدة أو الحرجة يتم تفكيكها إلى مستويات أدنى.

عدد العمليات إذا كان هناك عمليتان فقط في المستوى الأعلى، غالبًا لا تحتاجان إلى تفصيل إضافي. إذا تجاوز عدد العمليات **سبع عمليات** في مخطط واحد، يصبح المخطط معقدًا وصعب القراءة. لذلك يُفضل تقسيم العمليات الكبيرة إلى مخططات فرعية لتبسيط العرض.

- ❑ **المستوى الأعلى يحدد المدخلات والمخرجات.**
- ❑ **المستويات الأدنى تُظهر المصدر والوجهة لكل تدفق بيانات.**
- ❑ **العمق يعتمد على تعقيد النظام.**
- ❑ **العدد المثالي للعمليات في مخطط واحد يتراوح بين 3 إلى 7 عمليات.**

SUMMARY

- The Data Flow Diagram (DFD) is an essential tool for creating formal descriptions of business processes and data flows.
- Use cases record the input, transformation, and output of business processes.
- Eliciting scenario descriptions and modeling business processes are critically important skills for the systems analyst to master.

مخططات تدفق البيانات DFD تُعتبر أداة أساسية لإنشاء **وصف رسمي للعمليات التجارية وتدفق البيانات** داخل النظام. تساعد على فهم كيفية انتقال البيانات بين العمليات المختلفة، والمخازن، والكيانات الخارجية. تُستخدم بكثرة في تحليل الأنظمة لأنها تُظهر صورة واضحة ومبسطة لتدفق المعلومات.

حالات الاستخدام Use Cases تُسجل **المدخلات، التحويلات، والمخرجات** الخاصة بالعمليات التجارية. تُعتبر وسيلة نصية لوصف كيفية تفاعل المستخدم مع النظام. تُكمل مخططات DFDلأنها تُركز على **السيناريوهات العملية** التي يواجهها المستخدم.

دور محلل الأنظمة Systems Analyst من المهارات الحرجة التي يجب أن يتقنها محلل الأنظمة:

استخلاص أوصاف السيناريوهات Eliciting Scenario Descriptionsأي جمع المعلومات من المستخدمين وأصحاب المصلحة حول كيفية عمل النظام.

نمذجة العمليات التجارية Modeling Business Processes أي تحويل هذه الأوصاف إلى مخططات رسمية مثل DFDأو مخططات أخرى.

- DFDsتُظهر تدفق البيانات والعمليات بشكل رسومي.
- Use Casesتُسجل تفاصيل التفاعل بشكل نصي.
- الجمع بينهما يُعطي صورة متكاملة للنظام: نصية + رسومية.
- إتقان هذه الأدوات يُعتبر من أهم مهارات محلل الأنظمة لضمان بناء أنظمة دقيقة وفعّالة.

KEY DEFINITIONS

- A *data model* shows the people, places and things of interest to an organization and the relationships among them.
- The *logical data model* shows the organization of data without indicating how it is stored, created, or manipulated.
- A physical data model shows how the data will actually be stored in the database.
- Normalization is the process analysts use to check for data redundancy.

تعريفات أساسية في نمذجة البيانات Data Modeling

نموذج البيانات Data Model يُظهر **الأشخاص، الأماكن، والأشياء ذات الاهتمام للمؤسسة** والعلاقات فيما بينها. يُستخدم لفهم ما هي البيانات المهمة وكيف ترتبط ببعضها البعض.

النموذج المنطقي للبيانات Logical Data Model يُوضح **تنظيم البيانات** داخل النظام دون الإشارة إلى كيفية تخزينها أو إنشائها أو معالجتها. يُعتبر تمثيلًا مجردًا Abstract يركز على **المفاهيم والعلاقات** فقط.

النموذج الفيزيائي للبيانات Physical Data Model يُظهر **كيفية تخزين البيانات فعليًا في قاعدة البيانات**. يتضمن تفاصيل تقنية مثل:

- أسماء الجداول Tables
- الحقول Fields
- المفاتيح الأساسية والأجنبية Primary & Foreign Keys
- أنواع البيانات Data Types

التطبيع Normalization هو العملية التي يستخدمها المحللون للتحقق من **تكرار البيانات** Data Redundancy وتقليلها. يهدف إلى:

- تحسين تنظيم البيانات.
- ضمان الاتساق والدقة.
- تقليل التكرار غير الضروري.

ENTITY RELATIONSHIP MODEL

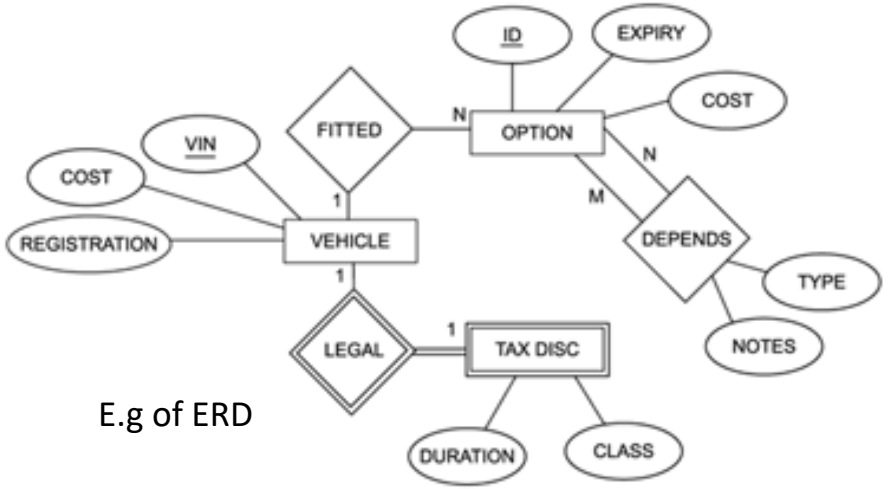
- **Entity-relationship model (ERM)** is an abstract and conceptual representation of data.
- Entity-relationship modeling is a database modeling method, used to produce a type of **conceptual schema** of a system.
- Diagrams created by this process are called **entity-relationship diagrams**, **ER diagrams**, or **ERDs**.

نموذج الكيان-العلاقة Entity-Relationship Model – ERM هو تمثيل **مجرد ومفاهيمي** **للبيانات** داخل النظام. يوضح الكيانات Entities ذات الاهتمام للمؤسسة والعلاقات Relationships التي تربط بينها.

نمذجة الكيان-العلاقة Entity-Relationship Modeling هي **طريقة لنمذجة قواعد البيانات** تُستخدم لإنتاج مخطط مفاهيمي Conceptual Schema للنظام. تساعد على فهم البيانات والعلاقات قبل الانتقال إلى التصميم الفيزيائي لقاعدة البيانات.

المخططات التي يتم إنشاؤها من هذه العملية تُسمى **مخططات الكيان-العلاقة** Entity-Relationship Diagrams أو اختصارًا: ER Diagrams / ERDs تُستخدم هذه المخططات لتوضيح:

- الكيانات (مثل: عميل، طلب، منتج).
- العلاقات (مثل: العميل يقدم طلبًا، الطلب يحتوي على منتجات).
- السمات (Attributes) الخاصة بكل كيان (مثل: اسم العميل، رقم الطلب).



E.g of ERD

WHAT IS AN ERD?

- A picture showing the information created, stored, and used by a business system.
- Entities generally represent people, places, and things of interest to the organization.
- Lines between entities show relationships between entities.

E-R MODEL CONSTRUCTS

- **Entity** - person, place, object, event, concept (often corresponds to a row in a table)
- **Attribute** - property or characteristic of an entity type (often corresponds to a field in a table)
- **Relationship** – link between entities (corresponds to primary key-foreign key equivalencies in related tables)

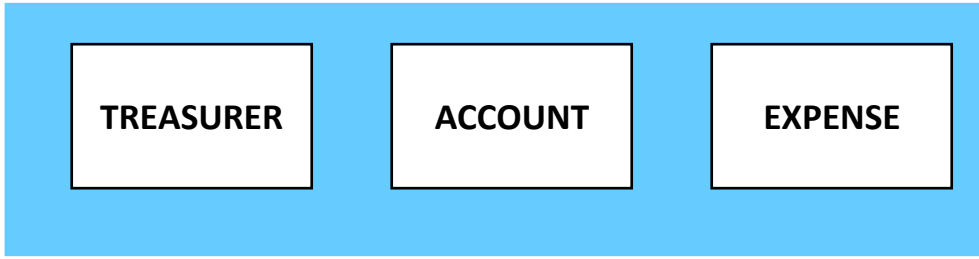
ERD ELEMENTS

	IDEF1X	Chen	Crow's Foot[a]
An ENTITY: ✔ Is a person, place, or thing ✔ Has a singular name spelled in all capital letters ✔ Has an identifier ✔ Should contain more than one instance of data	ENTITY-NAME Identifier	ENTITY-NAME	ENTITY-NAME *Identifier
An ATTRIBUTE: ✔ Is a property of an entity ✔ Should be used by at least one business process ✔ Is broken down to its most useful level of detail	ENTITY-NAME Attribute-name Attribute-name Attribute-name		ENTITY-NAME Attribute-name Attribute-name Attribute-name
A RELATIONSHIP: ✔ Shows the association between two entities ✔ Has modality (0,1) ✔ Has cardinality (1,M) ✔ Is described with a verb phrase	Relationship-name		Relationship-name

[a]This is the notation that will be used throughout the textbook

ENTITIES

- An entity is a person, place, object, event, or concept in the user environment about which the organization wishes to maintain data.
- E.g:
  - Person: EMPLOYEE, STUDENT, PATIENT
  - Place: STATE, REGION, COUNTRY, BRANCH
  - Object: MACHINE, BUILDING, AUTOMOBILE, PRODUCT
  - Event: SALE, REGISTRATION, RENEWAL
  - Concept: ACCT, COURSE, WORK CENTER



- Each entity in an E-R model is given a name.
- Since the name represent a class or set, it is singular.
- Also, since an entity is an object, we use simple noun to name entity type.
- CAPITAL LETTER-name is place inside a rectangle representing the entity.

WHAT SHOULD AN ENTITY BE?

- SHOULD BE:
  - An object that will have many instances in the database
  - An object that will be composed of multiple attributes
  - An object that we are trying to model
- SHOULD NOT BE:
  - A user of the database system
  - An output of the database system (e.g. a report)

ما يجب أن يكون الكيان Entity SHOULD BE

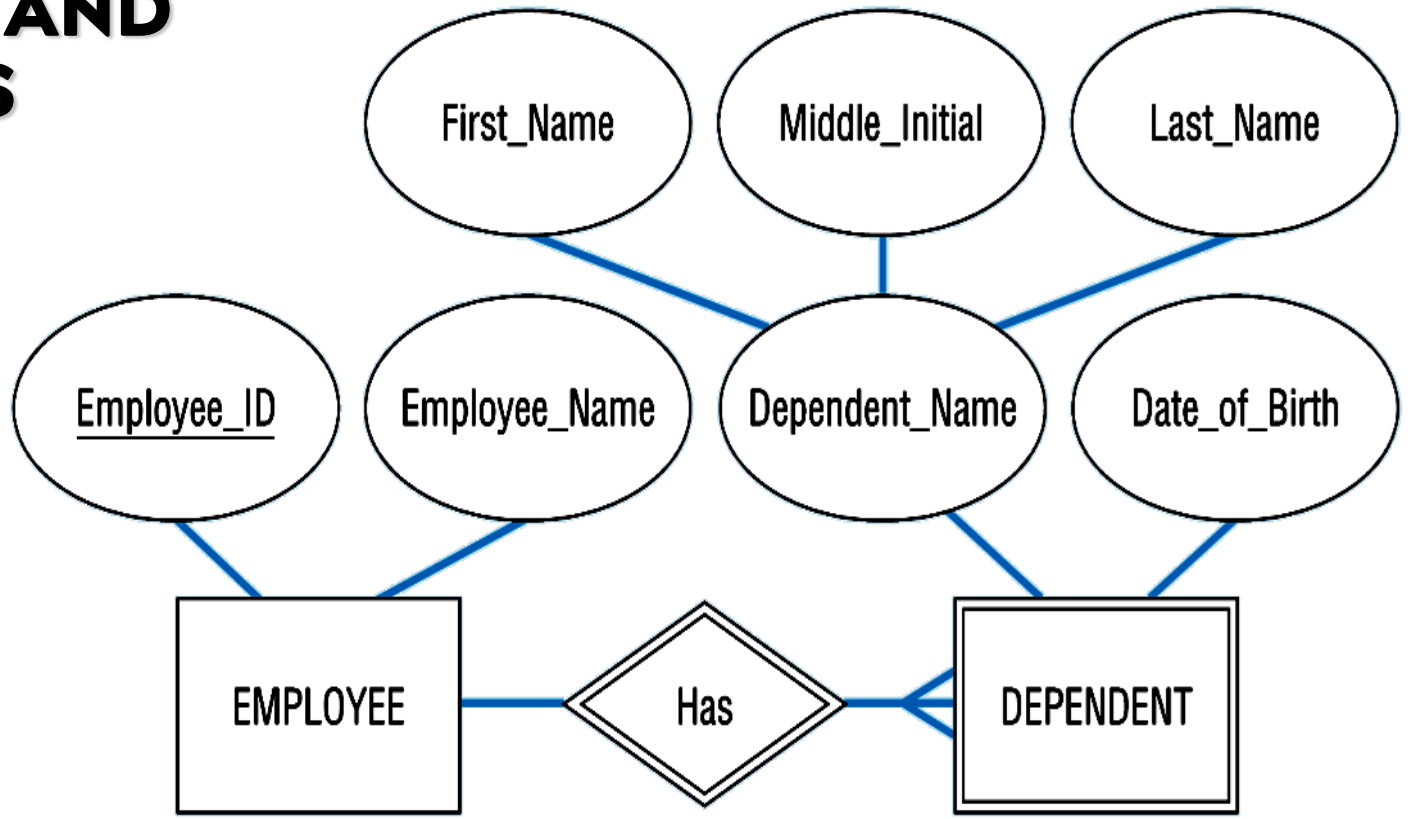
- **كائن له العديد من الحالات** Instances **في قاعدة البيانات**
  - مثال: عميل Customer يمكن أن يكون له آلاف السجلات.
- **كائن يتكون من عدة سمات** Attributes
  - مثال: عميل له سمات مثل: رقم العميل، الاسم، البريد الإلكتروني.
- **كائن نحاول نمذجته داخل النظام**
  - أي شيء مهم للمؤسسة ويجب تمثيله في قاعدة البيانات.

ما لا يجب أن يكون الكيان (Entity SHOULD NOT BE)

- **مستخدم نظام قاعدة البيانات**
  - مثل: "مدير قاعدة البيانات" أو "المبرمج" ← هؤلاء ليسوا كيانات، بل أشخاص يتعاملون مع النظام.
- **مخرجات النظام** Outputs مثل التقارير أو الشاشات
  - مثال: "تقرير المبيعات" أو "كشف الحساب" ← هذه ليست كيانات، بل نتائج يتم توليدها من البيانات.

STRONG VS. WEAK ENTITIES, AND IDENTIFYING RELATIONSHIPS

- Strong entities
  - exist independently of other types of entities
  - has its own unique identifier
  - represented with single-line rectangle
- Weak entity
  - dependent on a strong entity…cannot exist on its own
  - Does not have a unique identifier
  - represented with double-line rectangle
- Identifying relationship
  - links strong entities to weak entities
  - represented with double line diamond



Strong entity Identifying relationship Weak entity

- الكيان القوي = مستقل + مفتاح فريد + مستطيل خط واحد.
- الكيان الضعيف = يعتمد على قوي + لا يملك مفتاح فريد + مستطيل خط مزدوج.
- العلاقة المُعرِّفة = تربط القوي بالضعيف + معيّن خط مزدوج.

- *Cardinality* refers to the number of times instances in one entity can be related to instances in another entity
  - One instance in an entity refers to one and only one instance in the related entity (1:1)
  - One instance in an entity refers to one or more instances in the related entity (1:M)
  - One or more instances in an entity refer to one or more instances in the related entity (M:M)
- *Modality* refers to the minimum number of times that an instance in one entity can be related to an instance in another entity
  - One means that an instance in the related entity must exist for an instance in another entity to be valid
  - Zero means that no instance in the related entity is necessary for an instance in another entity to be valid

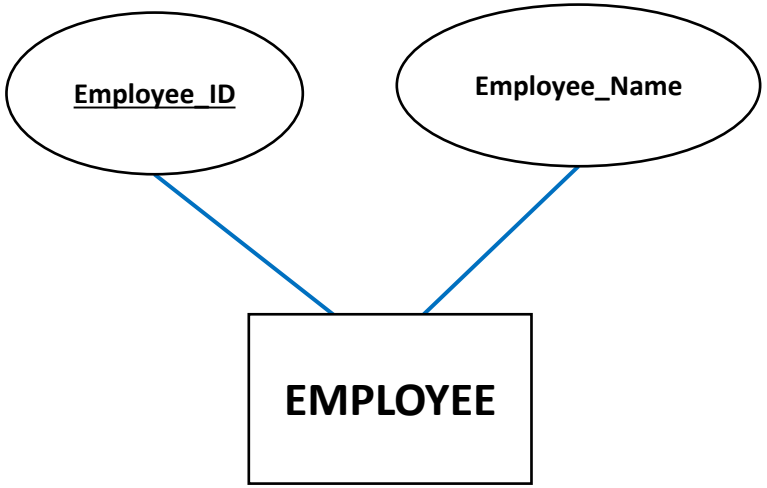
Cardinality تشير إلى **عدد المرات** التي يمكن أن ترتبط فيها حالات Instances من كيان واحد بحالات من كيان آخر.

- **1:1 (واحد لواحد):**
  - كل حالة في كيان ترتبط بحالة واحدة فقط في الكيان الآخر.
  - مثال: كل شخص لديه بطاقة هوية واحدة، وكل بطاقة هوية تخص شخصًا واحدًا.
- **1: M واحد لأكثر):**
  - كل حالة في كيان ترتبط بواحدة أو أكثر من الحالات في الكيان الآخر.
  - مثال: عميل واحد يمكن أن يقدم عدة طلبات، لكن كل طلب يخص عميلًا واحدًا فقط.
- **M:M) أكثر لأكثر):**
  - حالات متعددة في كيان ترتبط بحالات متعددة في كيان آخر.
  - مثال: طالب يمكن أن يسجل في عدة مقررات، وكل مقرر يمكن أن يضم عدة طلاب.

Modality تشير إلى **الحد الأدنى** لعدد المرات التي يمكن أن ترتبط فيها حالة من كيان بحالة من كيان آخر.

- **1 (إلزامي):**
  - يجب أن توجد حالة في الكيان المرتبط لكي تكون الحالة في الكيان الآخر صالحة.
  - مثال: لا يمكن أن يوجد طلب بدون عميل.
- **0 (اختياري):**
  - ليس من الضروري أن توجد حالة في الكيان المرتبط.
  - مثال: يمكن أن يوجد عميل في النظام حتى لو لم يقدم أي طلب بعد.

ATTRIBUTES

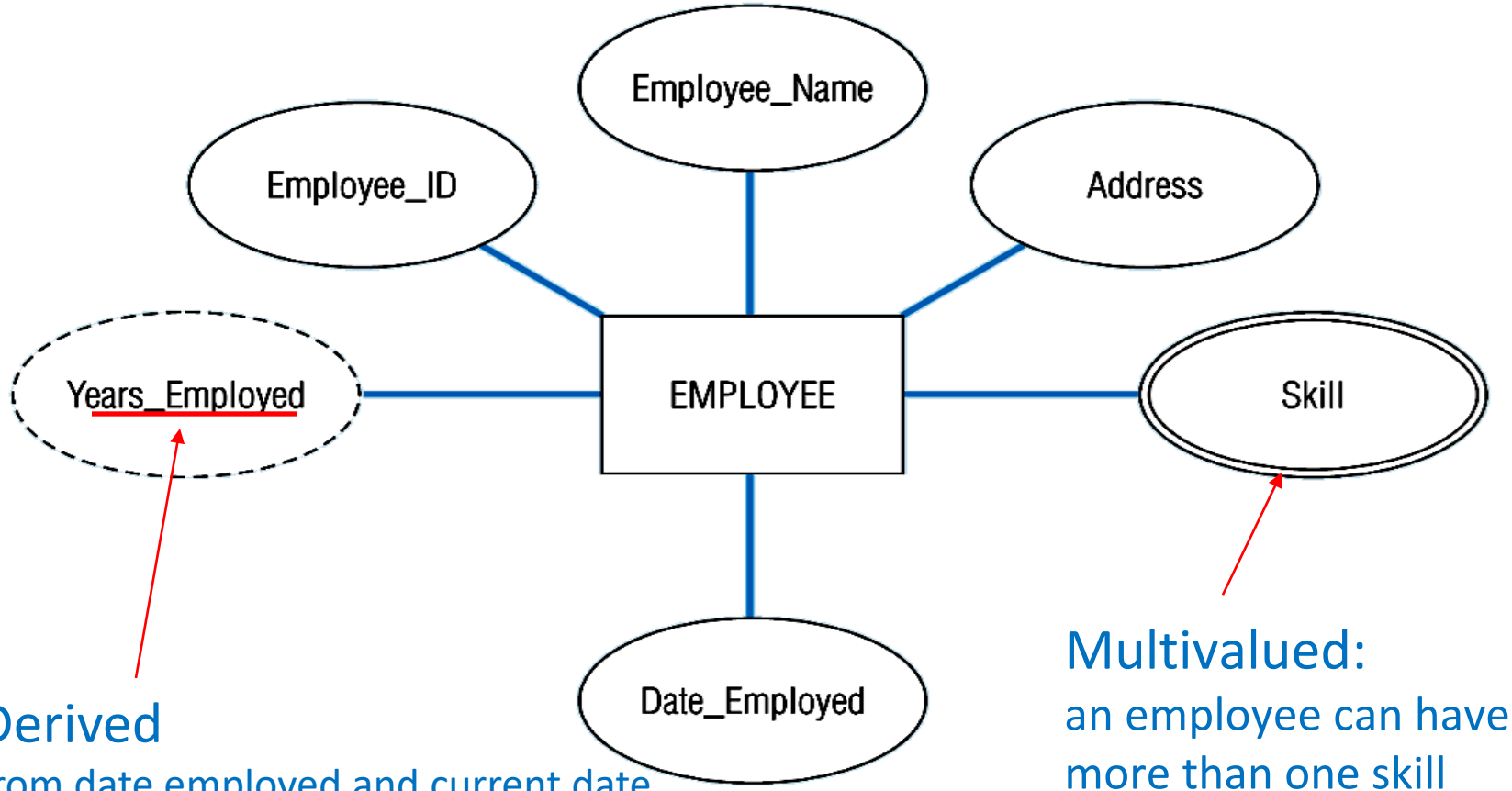


- Each entity type has a set of attributes associated with it.
- Attribute - property or characteristic of an entity type (relationship may also have attributes)
- Initial capital letter, followed by lowercase letters-nouns in naming an attribute.
- E.g:
  - EMPLOYEE: Employee_ID, Employee_Name, Address, Skill
  - STUDENT: Student_ID, Student_Name, Address, Phone_Number
  - AUTOMOBILE: Vehicle_ID, Color, Weight, Horsepower

IDENTIFIERS (KEYS)

- Every entity type must have an attribute or set of attributes that distinguishes one instance from other instances of the same type.
- Identifier (Key) - An attribute (or combination of attributes) that uniquely identifies individual instances of an entity type.

ENTITY WITH A MULTIVALUED ATTRIBUTE (SKILL) AND DERIVED ATTRIBUTE (YEARS_EMPLOYED)



Derived
from date employed and current date

Multivalued:
an employee can have more than one skill

Relationship

- Are the glue that hold together the various components of an E-R model.
- An association between the instances of one or more entity types that is of interest to the organization.
- An association means that event has occur or that there exist some natural linkage between entity instances.
- Labeled with verb phases.
  - Relationship between instances of two entity types.
  - The most common type of relationship encountered in data modeling.

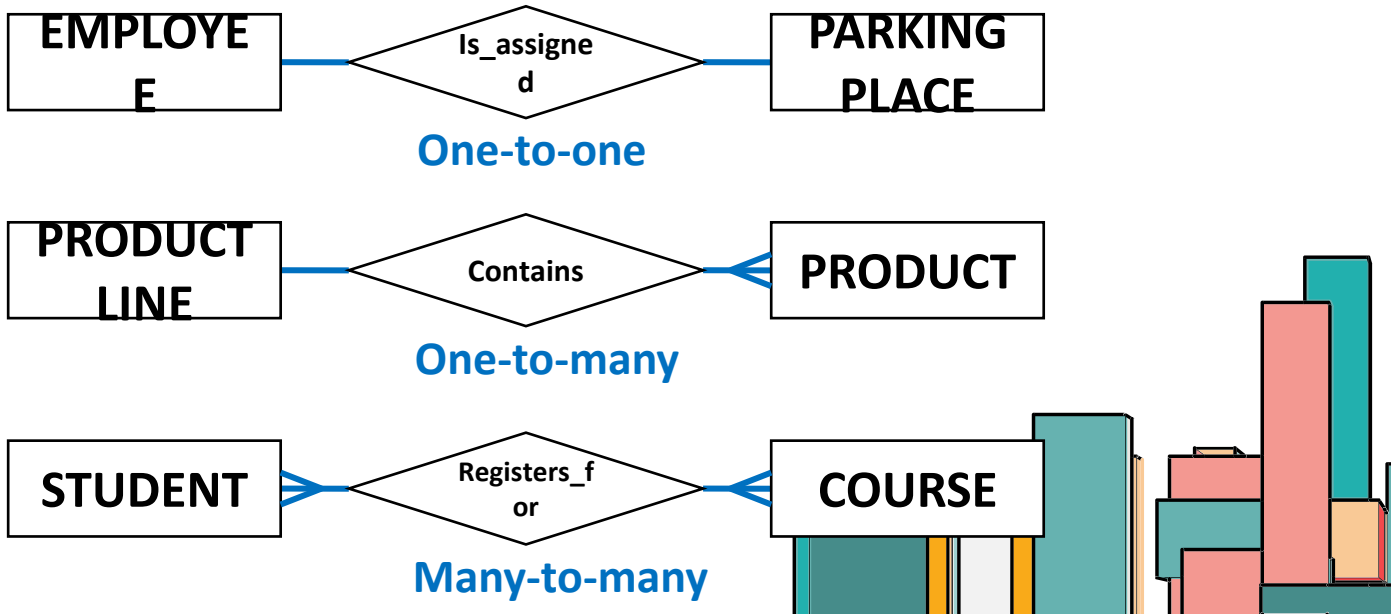
العلاقات هي **الغراء الذي يربط مكونات نموذج الكيان-العلاقة** معًا. تمثل **ارتباطًا بين الحالات** Instances من كيان واحد أو أكثر يكون ذا أهمية للمؤسسة. العلاقة تعني أن حدثًا قد وقع أو أن هناك ارتباطًا طبيعيًا بين الكيانات. تُسمى العلاقات عادةً باستخدام **عبارات فعلية** Verb Phrases لأنها تصف فعلًا أو ارتباطًا.

- مثال: **"يقدم"** places ، **"يحتوي"** ((contains، **"يسجل"** enrolls .

يمكن أن تكون العلاقة بين:

- كيان واحد وكيان آخر 1:1 أو M:1 .
- أو بين كيانات متعددة M:M .

تُظهر كيف تتفاعل الكيانات مع بعضها البعض داخل النظام.



RELATIONAL DATABASE MODEL (RDBM)

Requirement for Relation

- Relations have several properties that distinguish them from nonrelational tables:

  Every **relation** has a **unique name**.

  Every **attribute value is atomic** (not multivalued)

  Every **row is unique** (can't have two rows with exactly the same values for all their fields).

  Attributes in tables **have unique names**.

  The **order of the columns** *(left to right)* is **irrelevant**; can be interchange without changing the meaning or use of the relation.

  The **order of the rows** (top to bottom) is **irrelevant**; rows of a relation may be interchanged or stored in any sequence.

- A **data model** that **represents data** in the **form of tables** or **relations**.
- Relation - A named, two-dimensional table of data.
  - Each column in a relation corresponds to an attribute of the relation.
  - Each row in a relation corresponds to a record that contains data value for an entity.
- **Not all tables are relations.**

AUTHOR

au_id	au_lname	au_fname	address	city	state
172-32-1176	White	Johnson	10932 Bigge Rd.	Menlo Park	CA
213-46-8915	Green	Marjorie	309 63rd St. #411	Oakland	CA
238-95-7766	Carson	Cheryl	589 Darwin Ln.	Berkeley	CA
267-41-2394	O'Leary	Michael	22 Cleveland Av. #14	San Jose	CA
274-80-9391	Straight	Dean	5420 College Av.	Oakland	CA
341-22-1782	Smith	Meander	10 Mississippi Dr.	Lawrence	KS
409-56-7008	Bennet	Abraham	6223 Bateman St.	Berkeley	CA
427-17-2319	Dull	Ann	3410 Blonde St.	Palo Alto	CA
472-27-2349	Gringlesby	Burt	PO Box 792	Covelo	CA
486-29-1786	Locksley	Charlene	18 Broadway Av.	San Francisco	CA

TITLE

title_id	title	type	price	pub_id
BU1032	The Busy Executive's Database Guide	business	19.99	1389
BU1111	Cooking with Computers	business	11.95	1389
BU2075	You Can Combat Computer Stress!	business	2.99	736
BU7832	Straight Talk About Computers	business	19.99	1389
MC2222	Silicon Valley Gastronomic Treats	mod_cook	19.99	877
MC3021	The Gourmet Microwave	mod_cook	2.99	877
MC3026	The Psychology of Computer Cooking	UNDECIDED		877
PC1035	But Is It User Friendly?	popular_comp	22.95	1389
PC8888	Secrets of Silicon Valley	popular_comp	20	1389
PC9999	Net Etiquette	popular_comp		1389
PS2091	Is Anger the Enemy?	psychology	10.95	736

PUBLISHER

pub_id	pub_name	city
736	New Moon Books	Boston
877	Binnet & Hardley	Washington
1389	Algodata Infosystems	Berkeley
1622	Five Lakes Publishing	Chicago
1756	Ramona Publishers	Dallas
9901	GGG&G	München
9952	Scootney Books	New York
9999	Lucerne Publishing	Paris

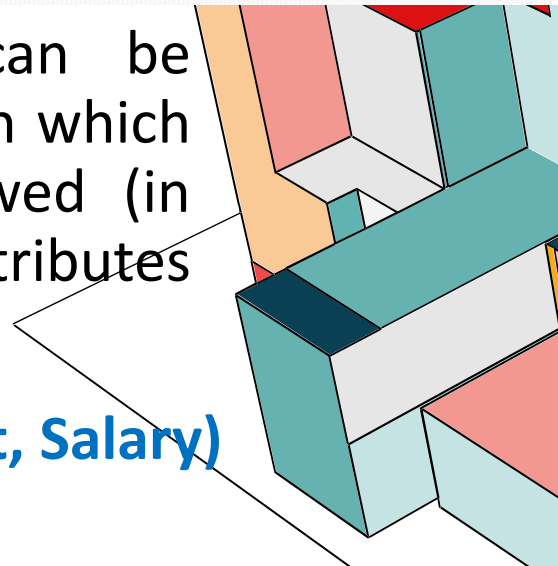
AUTHOR_TITLE

au_id	title_id
172-32-1176	PS3333
213-46-8915	BU1032
213-46-8915	BU2075
238-95-7766	PC1035
267-41-2394	BU1111
267-41-2394	TC7777
274-80-9391	BU7832
409-56-7008	BU1032
427-17-2319	PC8888
472-27-2349	TC7777

- The *structure* of the relation can be expressed by a shorthand notation in which the name of the relation is followed (in parentheses) by the names of the attributes in the relation.

EMPLOYEE(Employee_ID, Name, Dept, Salary)

- The primary attribute is underlined.



CORRESPONDENCE WITH ER MODEL

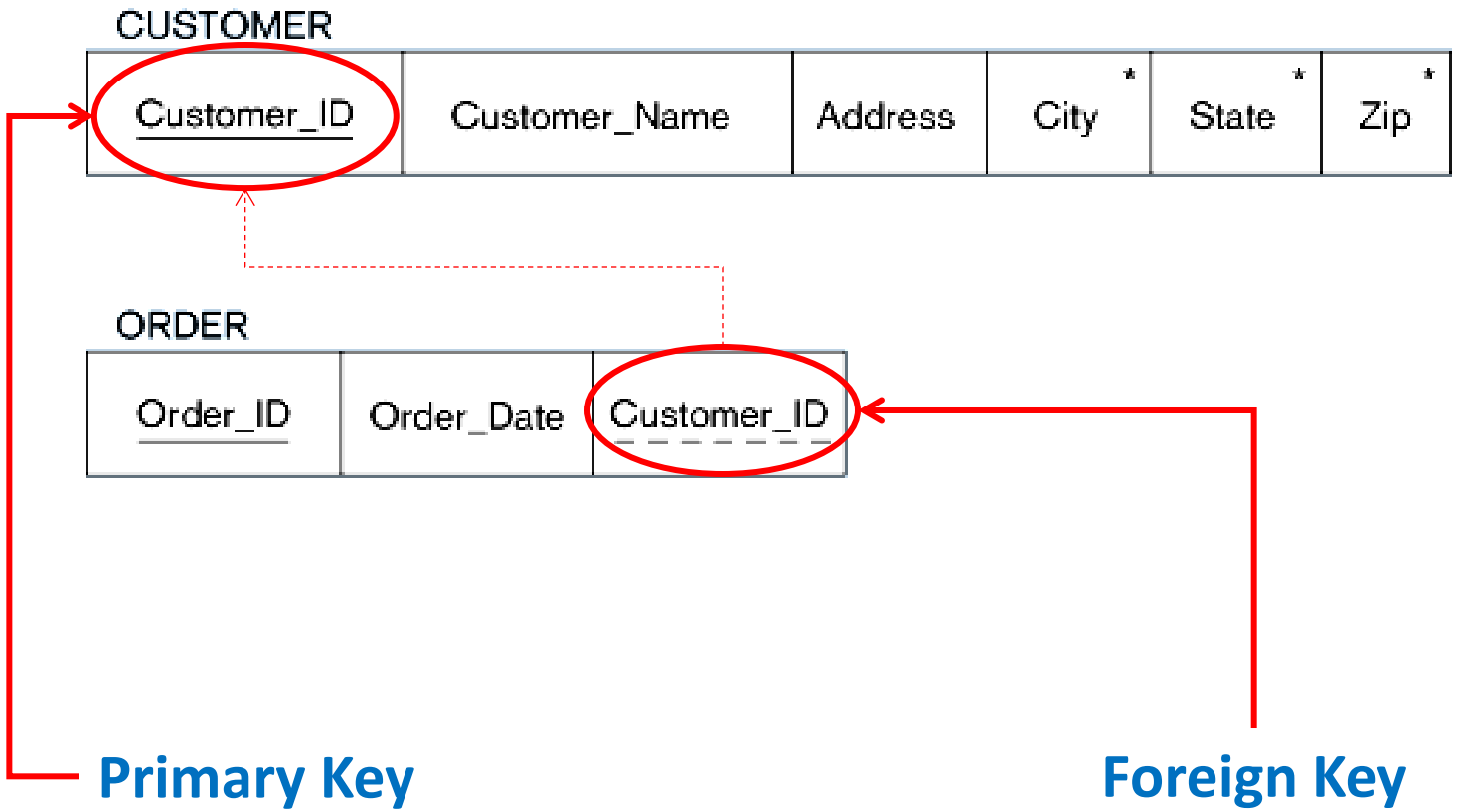
- Relations correspond with entity types and with many-to-many relationship types
- Rows correspond with entity instances and with many-to-many relationship instances
- Columns correspond with attributes

NOTE: The word ***relation*** (in relational database) is NOT the same the word ***relationship*** (in ER model)

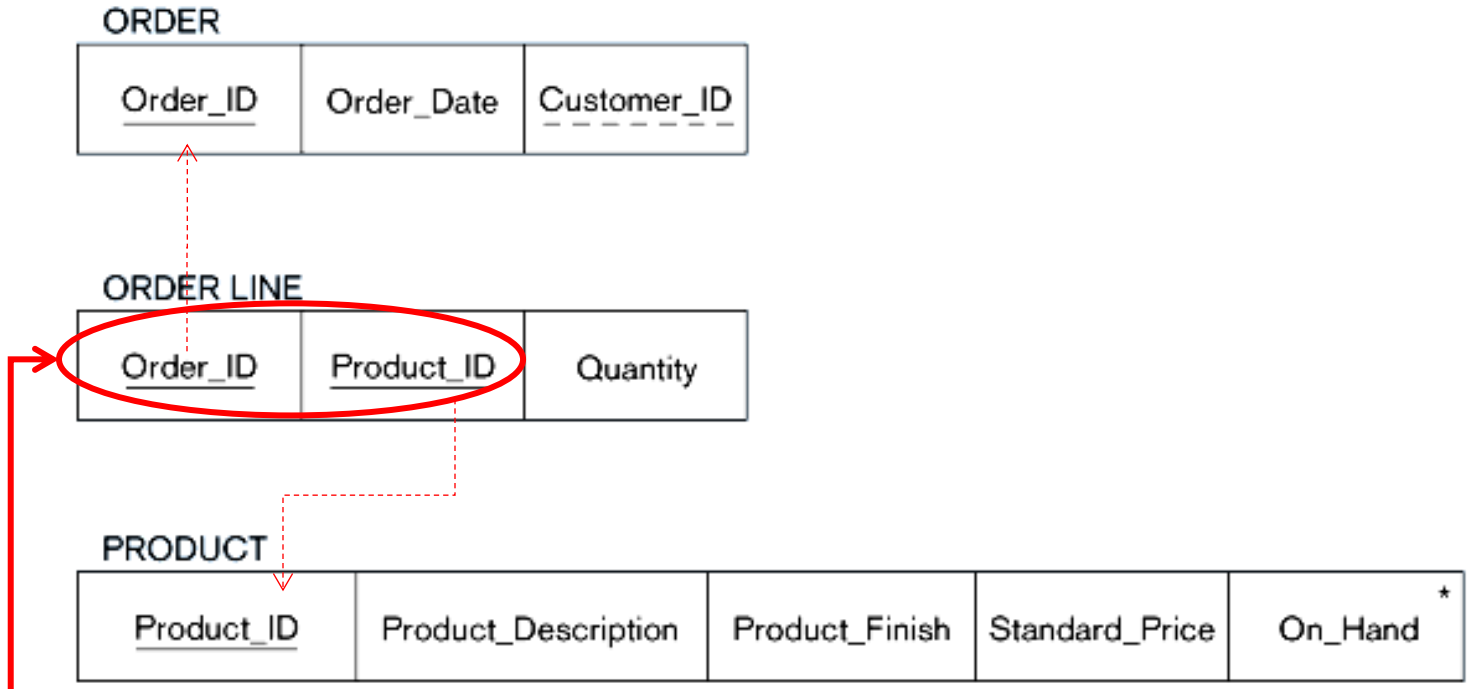
KEY FIELDS

- Keys are special fields that serve two main purposes:
  - ***Primary key*** is a name(s) of attribute(s) that form the unique identifier for each row of relation. Examples include employee numbers, social security numbers, etc.
  - ***Foreign key*** is an attribute that appears as a nonkey attribute in one relation and as a primary key attribute in another relation.
- Keys can be ***simple*** (a single field) or ***composite*** (more than one field)
- Keys usually are used as indexes to speed up the response to user queries.

SCHEMA FOR RELATIONS



SCHEMA FOR RELATIONS

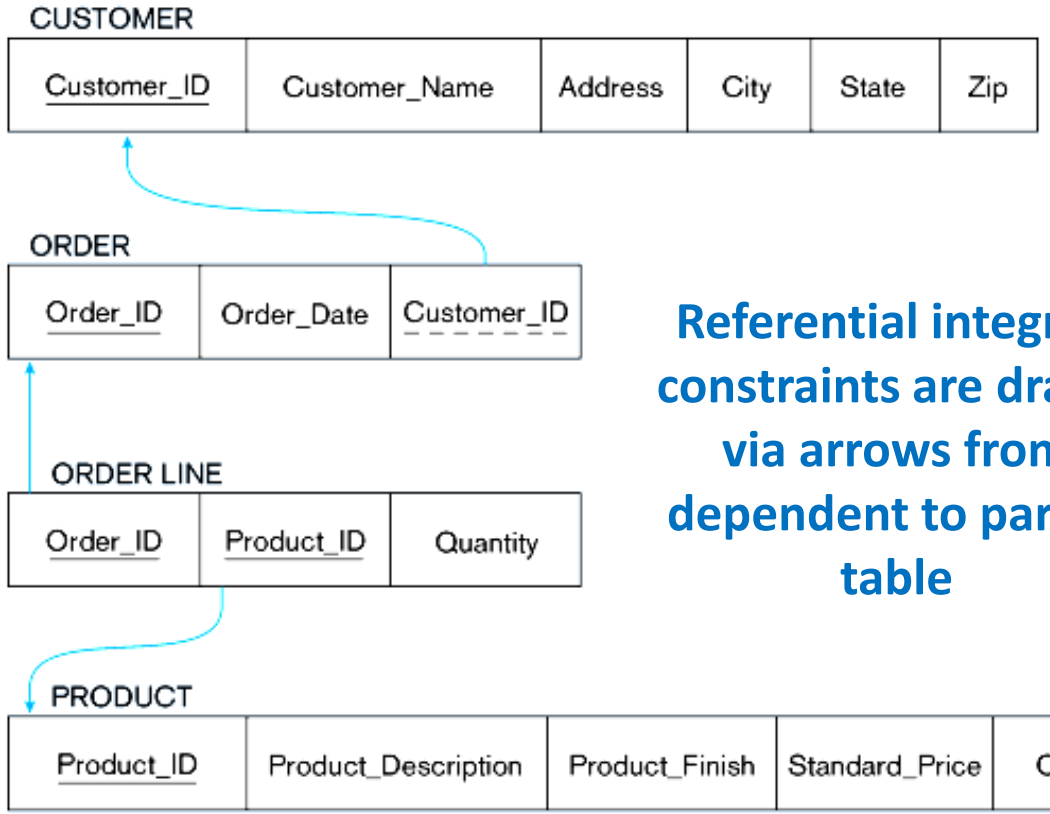


Combined, these are a **composite primary key** (uniquely identifies the order line)…individually they are **foreign keys**

INTEGRITY CONSTRAINTS

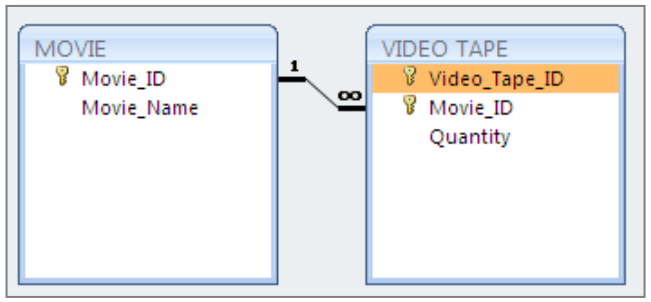
- Referential Integrity – specify that the value of an attribute in one relation depends on the value of the same attribute in another relation.
- Rule that states that any foreign key value (on the relation of the many side) MUST match a primary key value in the relation of the one side.
- Entity Integrity - No primary key attribute may be null. All primary key fields **MUST** have data.

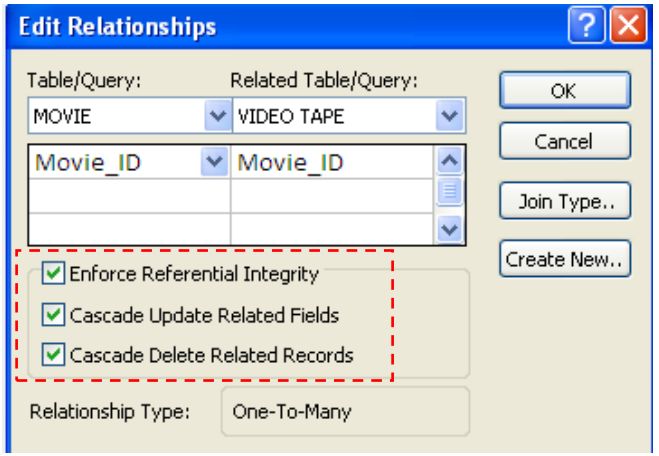
REFERENTIAL INTEGRITY CONSTRAINTS



Referential integrity constraints are drawn via arrows from dependent to parent table

REFERENTIAL INTEGRITY





- Consider we have 2 tables: MOVIE and VIDEO TAPE. The VIDEO TAPE table has a foreign key attribute entitled Movie_ID which points to the record for that Movie_ID in the MOVIE table.
- Enforces three rules:
  - We may not add a record to the VIDEO TAPE table unless the attribute points to a valid record in MOVIE table.
  - If the primary key for a record in the MOVIE table changes, all corresponding records in the VIDEO TAPE table must be modified using a cascading update.
  - If a record in the MOVIE table is deleted, all corresponding records in the VIDEO TAPE table must be deleted using a cascading delete.

مثال MOVIE و VIDEO_TAPE

الجداول

- MOVIE
  - يحتوي على معلومات عن الأفلام.
  - المفتاح الأساسي: Movie_ID.
- VIDEO_TAPE
  - يحتوي على نسخ الفيديو للأفلام.
  - يحتوي على مفتاح أجنبي: Movie_ID يشير إلى الجدول MOVIE.

القواعد الثلاثة التي يفرضها المفتاح الأجنبي

1. **عدم إضافة سجل غير صالح** Insert Rule
   1. لا يمكن إضافة سجل في جدول VIDEO_TAPE إلا إذا كان Movie_ID يشير إلى سجل موجود في جدول MOVIE.
   2. هذا يمنع وجود أشرطة فيديو مرتبطة بأفلام غير موجودة.
2. **التحديث المتسلسل** Cascading Update
   1. إذا تغير المفتاح الأساسي Movie_ID في جدول MOVIE، يجب تعديل جميع السجلات المرتبطة في جدول VIDEO_TAPE تلقائيًا.
   2. هذا يحافظ على الاتساق بين الجداول.
3. **الحذف المتسلسل** Cascading Delete
   1. إذا تم حذف سجل من جدول MOVIE، يجب حذف جميع السجلات المرتبطة به في جدول VIDEO_TAPE تلقائيًا.
   2. هذا يمنع وجود سجلات orphaned يتيمة في جدول الفيديو.

TRANSFORMING ER DIAGRAMS INTO RELATIONS

- Represent entities
  - Each entity type in E-R diagram is represented as a relation in the relational data model.
  - Identifier (entity type) -> primary key (relation)
  - Attribute(s) (entity type) -> nonkey attribute (s) (relation)
- Represent relationships
  - Each relationship in an E-R diagram must be represented in the relational model.
  - How to represent relationship is depends on its nature.
- Normalize the relations
  - The relations that are created in step 1 and 2 may have unnecessary redundancy and may be subject to anomalies when they are updated.
  - Normalize the relations to make it well-structured.

تمثيل الكيانات Represent Entities كل **كيان** Entity Type في مخطط ERيُمثل كـ **علاقة** Relation/Table في النموذج العلائقي. **المعرّف** Identifier في الكيان يصبح **المفتاح الأساسي** Primary Key في الجدول. **السمات** Attributes في الكيان تصبح **حقول غير مفتاحية** Non-key Attributes في الجدول.

مثال: كيان Customer → جدول Customer

Primary Key: Customer_ID

Non-key Attributes: Name, Email, Phone

تمثيل العلاقات Represent Relationships كل **علاقة** Relationship في مخطط ER

يجب أن تُترجم إلى النموذج العلائقي. طريقة التمثيل تعتمد على طبيعة العلاقة:

- ❑ **1:1 واحد لواحد :** غالبًا يُدمج المفتاح الأساسي لأحد الكيانات في الآخر كمفتاح أجنبي.
- ❑ **1:M واحد لأكثر :** المفتاح الأساسي للكيان "الواحد" يصبح مفتاحًا أجنبيًا في الكيان "الأكثر".
- ❑ **M:M أكثر لأكثر :** يتم إنشاء **جدول وسيط** Intersection Table
  - يحتوي على المفاتيح الأساسية للكيانين كـ **مفاتيح أجنبية**.

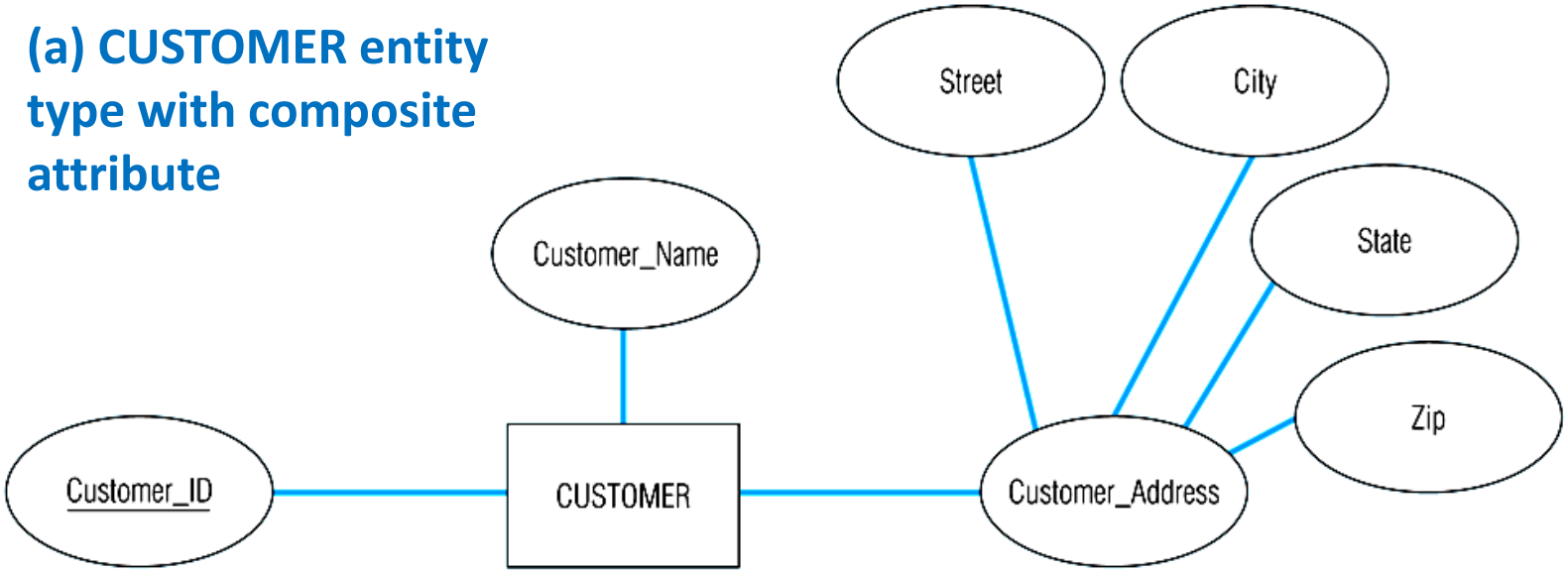
التطبيع Normalization الجداول الناتجة من الخطوتين السابقتين قد تحتوي على **تكرار غير ضروري** Redundancy أو قد تكون عرضة لمشاكل عند التحديث Update Anomalies لذلك يتم تطبيق **التطبيع** Normalization لضمان أن العلاقات:

- منظمة بشكل جيد Well-structured
- خالية من التكرار غير الضروري.
- تحافظ على الاتساق والدقة.

- ❖ **الكيانات** → **جداول** (معرّف = مفتاح أساسي، سمات = حقول).
- ❖ **العلاقات** → **مفاتيح أجنبية أو جداول وسيطة** حسب طبيعة العلاقة.
- ❖ **التطبيع** → **إزالة التكرار وضمان الاتساق**.

TRANSFORMING ER DIAGRAMS INTO RELATIONS

(a) CUSTOMER entity type with composite attribute



(b) CUSTOMER relation with address detail

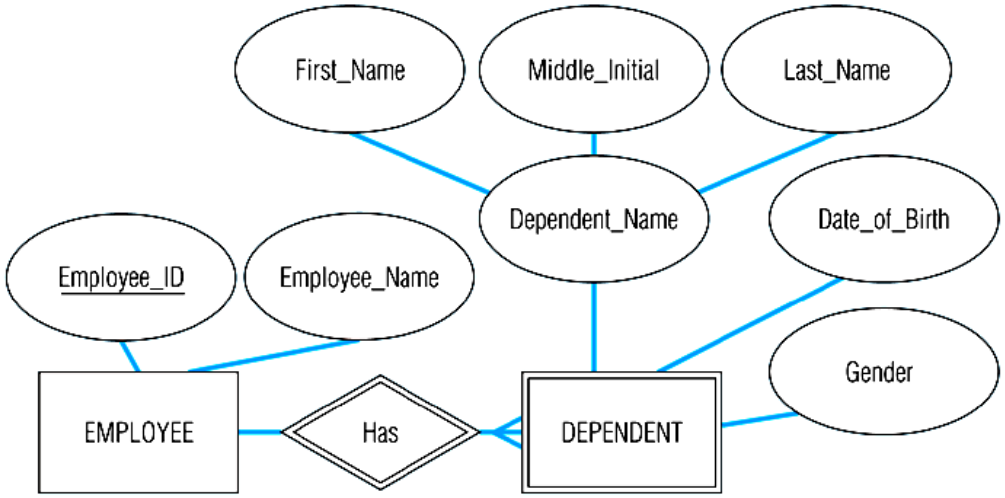
CUSTOMER

Customer_ID	Customer_Name	Street	City	State	Zip

TRANSFORMING ER DIAGRAMS INTO RELATIONS

Mapping Weak Entities

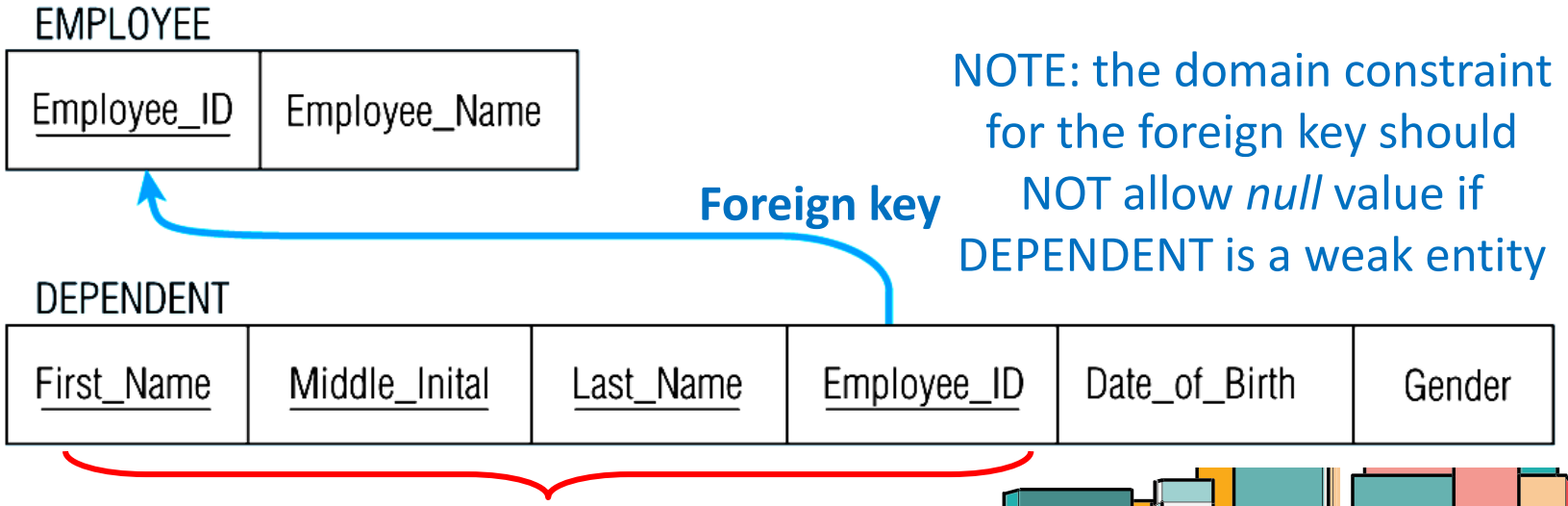
- Becomes a separate relation with a foreign key taken from the superior entity
- Primary key composed of:
  - Partial identifier of weak entity
  - Primary key of identifying relation (strong entity)



(a) Weak entity DEPENDENT

EXAMPLE OF MAPPING A WEAK ENTITY

RELATIONS RESULTING FROM WEAK ENTITY



NOTE: the domain constraint for the foreign key should NOT allow *null* value if DEPENDENT is a weak entity

TRANSFORMING ER DIAGRAMS INTO RELATIONS

Represent Relationships

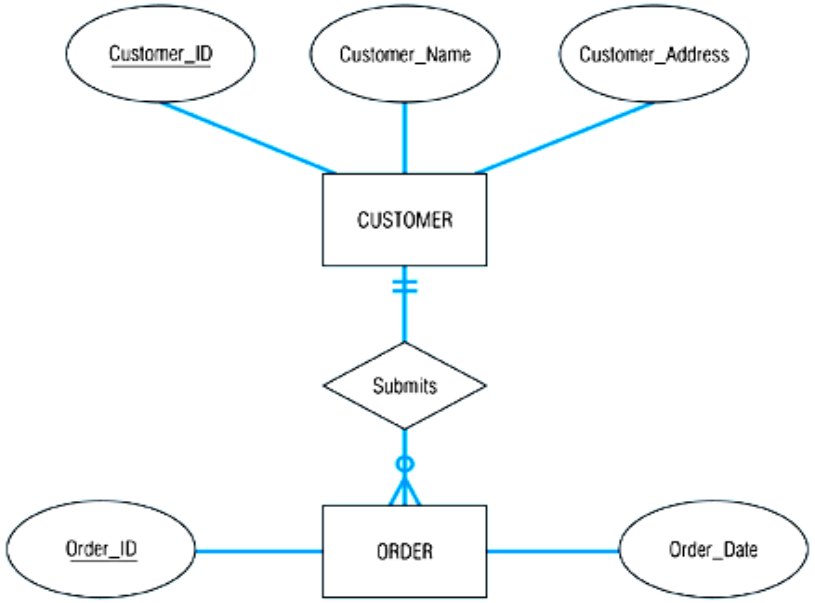
- The procedure depends on both the degree of the relationship – unary, binary, ternary- and cardinalities of the relationship.

Mapping Binary Relationships

- One-to-Many (1:N)
  - Primary key on the one side becomes a foreign key on the many side
- Many-to-Many (M:N)
  - Create a ***new relation*** with the primary keys of the two entities as its primary key
- One-to-One (1:1)
  - Primary key on the mandatory side becomes a foreign key on the optional side

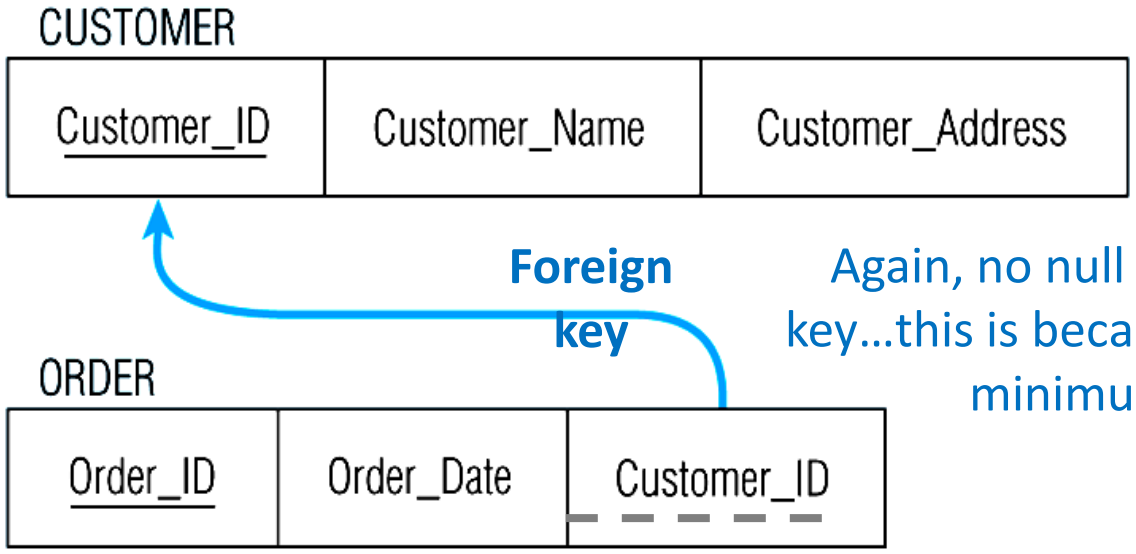
BINARY 1:N RELATIONSHIP

(a) Relationship between customers and orders



Note the mandatory one

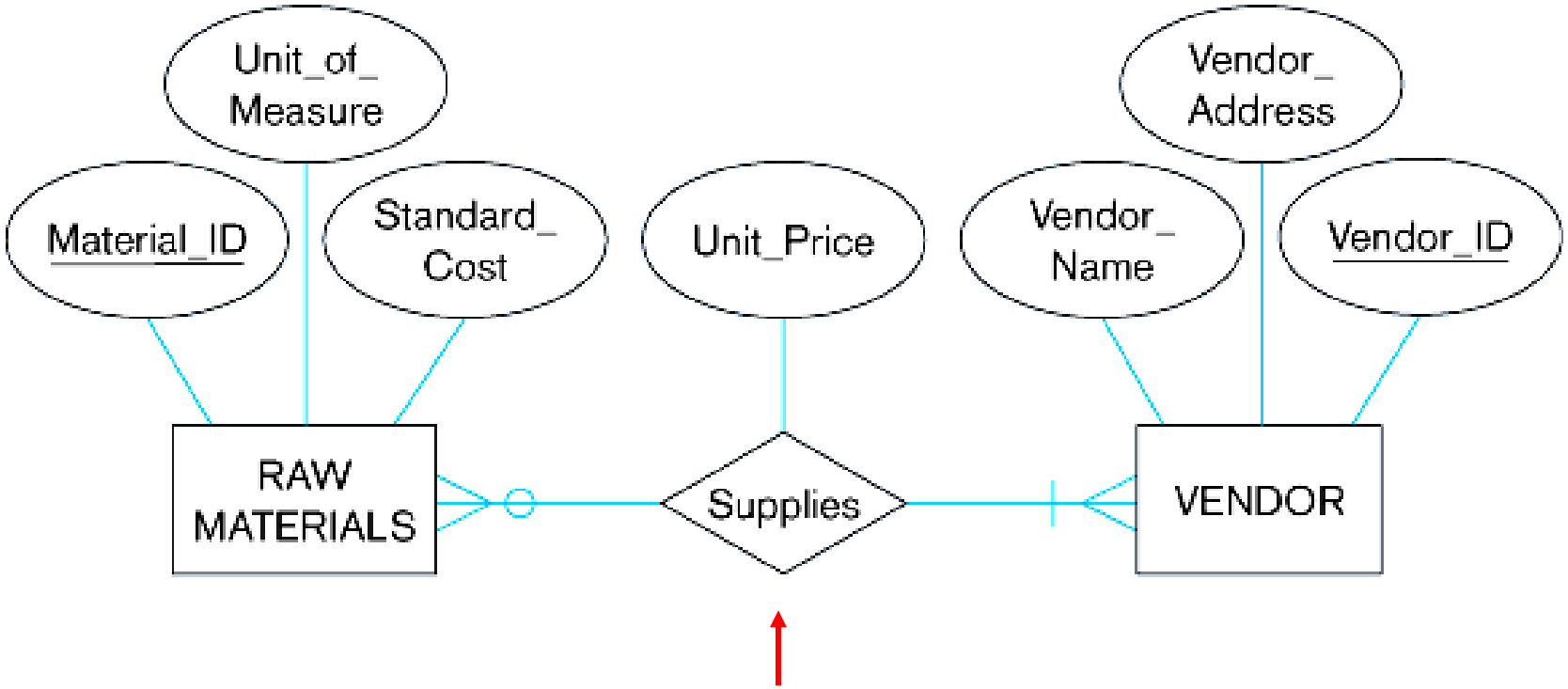
MAPPING THE RELATIONSHIP



Again, no null value in the foreign key…this is because of the mandatory minimum cardinality

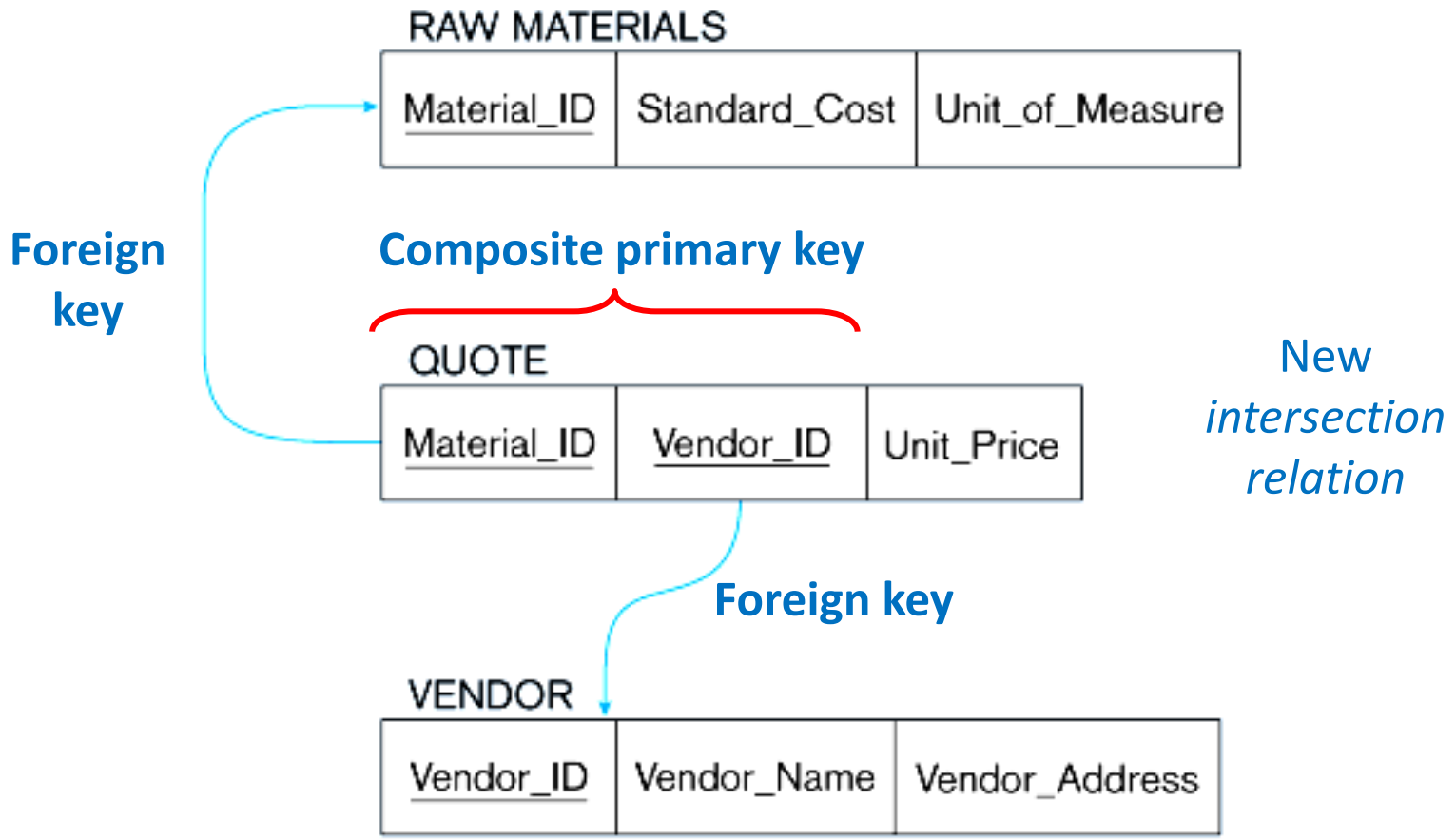
BINARY M:N RELATIONSHIP

(a) ER diagram (M:N)

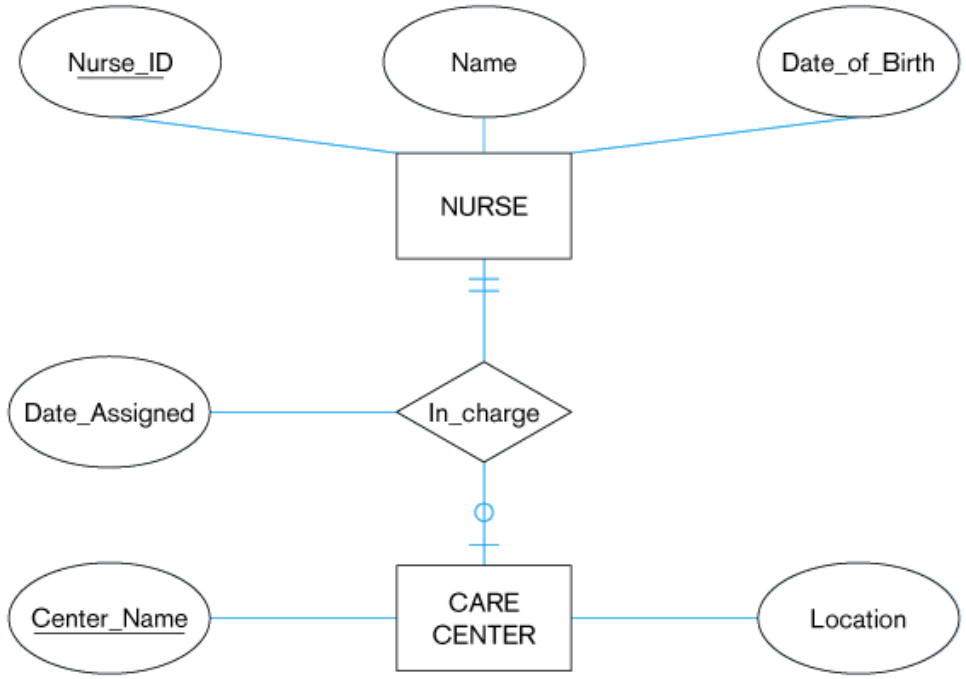


The Supplies relationship will need to become a separate relation

THREE RESULTING RELATIONS



BINARY 1:1 RELATIONSHIP



(a) Binary 1:1 relationship

RESULTING RELATIONS

NURSE

Nurse_ID	Name	Date_of_Birth

CARE CENTER

Center_Name	Location	Nurse_in_Charge	Date_Assigned

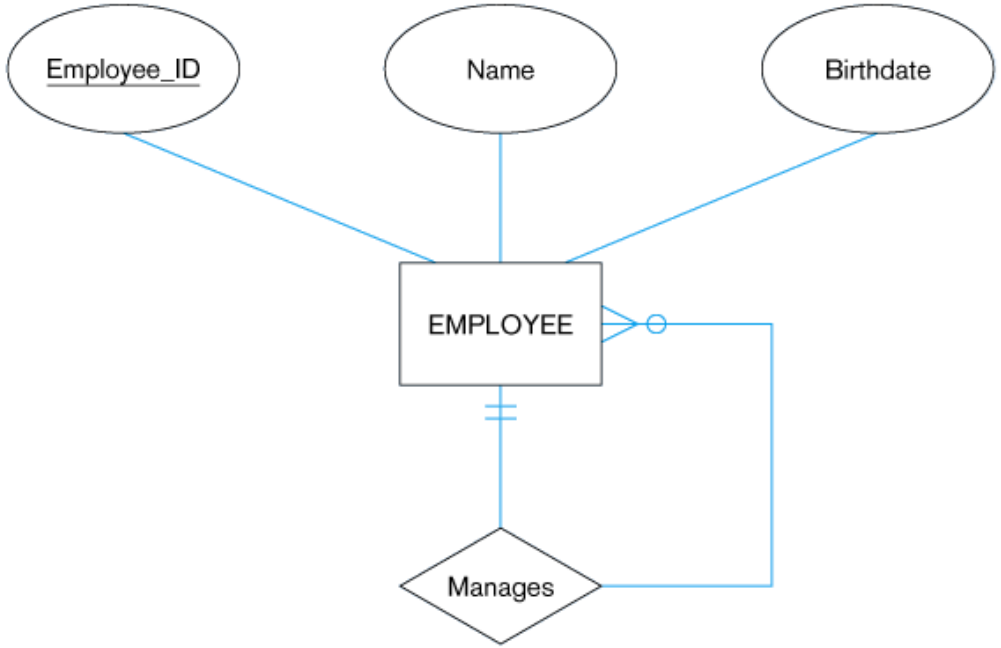
TRANSFORMING ER DIAGRAMS INTO RELATIONS

Mapping Unary Relationships

- One-to-Many (1:N)
  - Recursive foreign key in the same relation
- Many-to-Many (M:N)
  - Two relations:
  - One for the entity type
  - One for an associative relation in which the primary key has two attributes, both taken from the primary key of the entity

UNARY 1:N RELATIONSHIP

(a) EMPLOYEE entity with Manages relationship

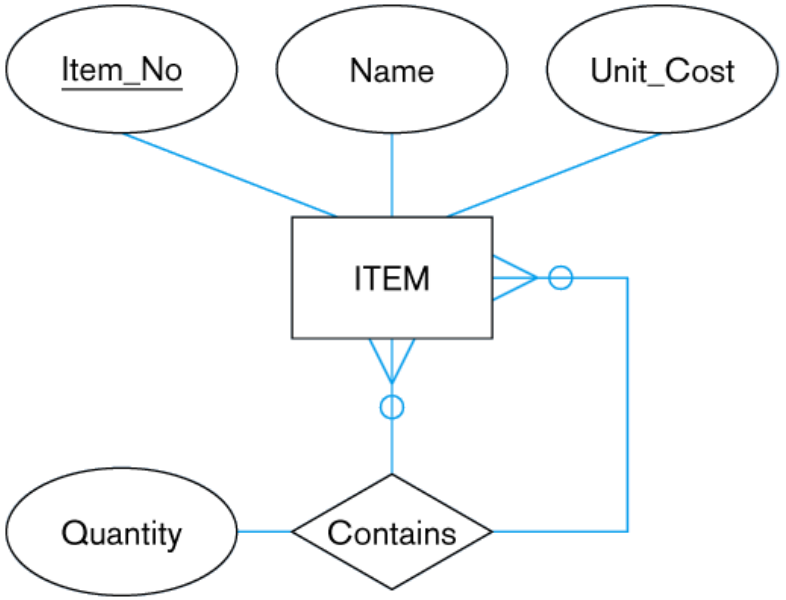


EMPLOYEE

Employee_ID	Name	Birthdate	Manager_ID

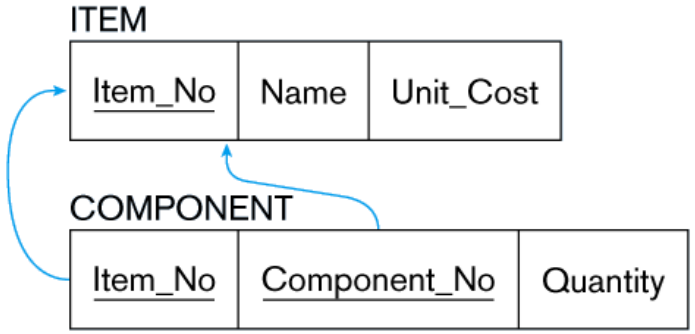
(b) EMPLOYEE relation with recursive foreign key

UNARY M:N RELATIONSHIP



(a) Bill-of-materials relationships (M:N)

(b) ITEM and COMPONENT relations



SUMMARY OF CONCEPTUAL DATA MODELING WITH E-R DIAGRAMS

- The purpose of E-R diagramming is to capture the richest possible understanding of the meaning of data necessary for an information system or organization.
- Besides aspects shown in this chapter, there are many other semantics about data that E-R diagram can represent and some of these are more advance capabilities.

ERD BASICS

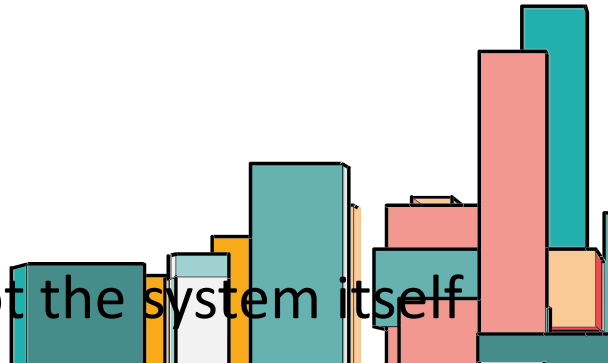
- Drawing the ERD is an iterative process of trial and revision
- ERDs can become quite complex

STEPS IN BUILDING ERDS

- Identify the entities
- Add appropriate attributes for each entity
- Draw the relationships that connect associated entities

ERD BUILDING TIPS

- Data stores of the DFD should correspond to entities
- Only include entities with more than one instance of information
- Don't include entities associated with implementation of the system, not the system itself

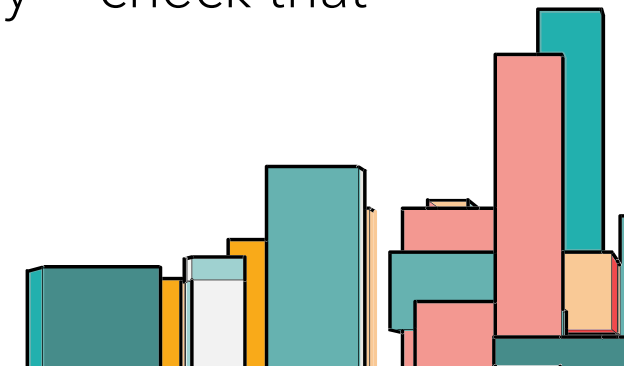


DESIGN GUIDELINES

- Best practices rather than rules
- Entities should have many occurrences
- Avoid unnecessary attributes
- Clearly label all components
- Apply correct cardinality and modality
- Break attributes into lowest level needed
- Labels should reflect common business terms
- Assumptions should be clearly stated

BALANCING ERDS WITH DFDS

- All analysis activities are interrelated
- Process models contain two data components
  - Data flows and data stores
- The DFD data components need to balance the ERD's data stores (entities) and data elements (attributes)
- Many CASE tools provide features to check for imbalance
- Check that all data stores and elements correspond between models
- Do not follow thoughtlessly -- check that the models make sense!



SUMMARY

- The ERD is the most common technique for drawing data models. The building blocks of the ERD are:
  - Entities describe people, places, or things
  - Attributes capture information about the entity
  - Relationships associate data across entities
- Intersection, dependent, and independent entities must be recognized.
- The ERD must be balanced with the DFD.

APPENDIXES

ASSIGNMENT 1

- The previous questions helps in defining the scope of that project. Answers helps in having a clear consensus about if this production project could be conducted or not.
- If there some technical, legal, expertise- related, political, or other could arise, and couldn't be mitigated, then the production project is not feasible.
- If the dedicated budget was not reasonable at all then, Boeing will notify Libyan Airline that this plane couldn't be produced and delivered to Libya.
- In contrary, if the project shows feasibility then, the rest of information will help in defining the scope of the project.