**Introduction to Data Modelling**

1. **The Entity Relationship Model 实体关系模型**
2. **Basic ER modeling concepts**

* **Entity: Real world object distinguishable from other objects. An entity is described (in DB) using a set of attributes.**

**实体:真实世界的对象，区别于其他对象。使用一组属性描述一个实体(在DB中)。**

* **Entity Set: A collection of entities of the same type (e.g. all employees)**

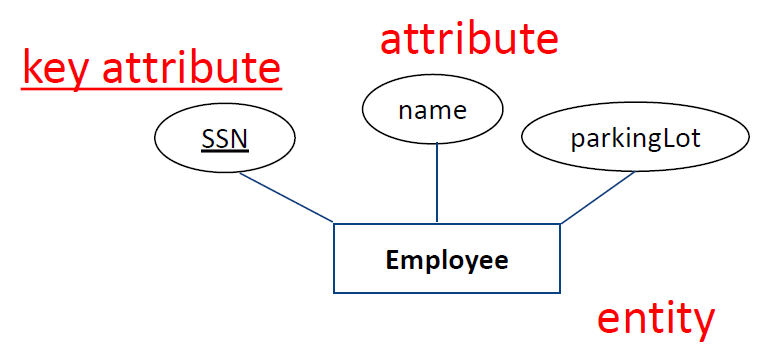
**实体集:同一类型的实体的集合(如所有雇员)**

* **All entities in an entity set have the same set of attributes**

**实体集中的所有实体都具有相同的属性集**

* **Each entity has a unique key (underlined)**

**每个实体都有一个唯一的键(带下划线)**

****

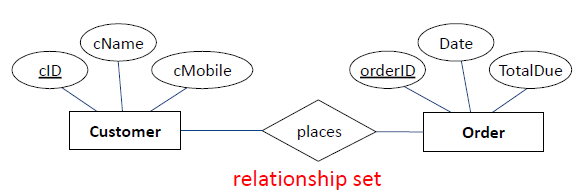
* **Relationship: Association among two or more entities.**

**关系:两个或多个实体之间的关联。**

* **Example: John places a Pizza order.**
* **Relationship Set: Collection of relationships of the same type.**

**关系集:同一类型的关系的集合。**

* **Example: Customers place orders.**

****

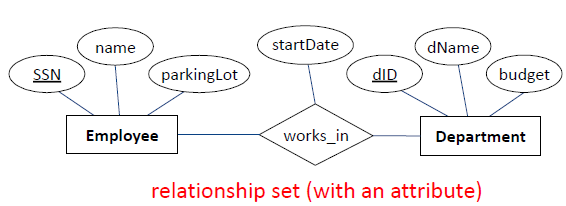
* **Relationship: Relationships can have their own attributes.**

**关系:关系可以有自己的属性。**

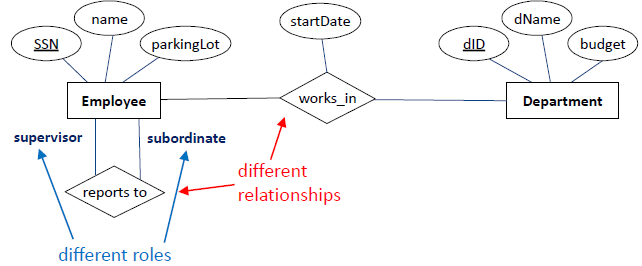
* **Example: John works in the Pharmacy department starting 1 July 2021.**
* **Relationship Set: Collection of relationships of the same type.**

**关系集:同一类型的关系的集合。**

* **Example: Employees work in departments.**

****

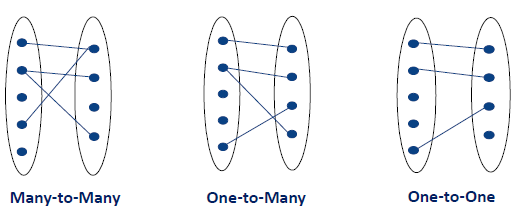
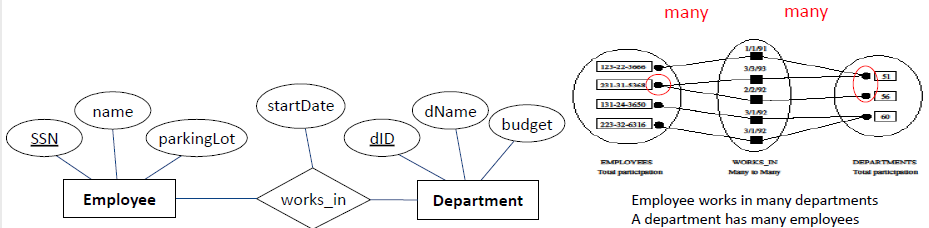
* **Same entity set can participate in:**
* **different relationship sets, or even**
* **different roles” in the same set**

****

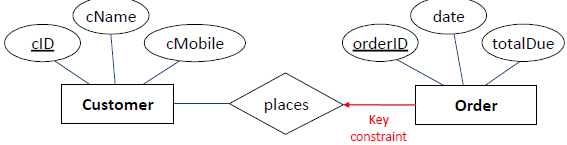
1. **Constraints**

* **Key constraints determine the number of objects taking part in the relationship set (how many from each side)**

**关键约束确定参与关系集中的对象的数量(每边有多少个)**

* **Types of key constraints:**
* ****
* **Many-to-Many**
* **An employee can work in many departments; a department can have many employees. Many is represented by a line.**
* ****
* **Key Constraints: One to Many**
* **One to many constrains one entity set to have a single entity per a relationship. An entity of that set can never participate in two relationships of the same relationship set. This is called a key constraint and is represented by an arrow.**

**一对多约束将一个实体设置为每个关系有一个实体。这个集合的一个实体永远不能参与同一关系集合的两个关系。这被称为键约束，用箭头表示。**

* **Example:**
* **A customer can place many orders**
* **Each order belongs to at most one customer.**
* **This is the key constraint on Places**
* ****
* **Participation Constraints**
* **Participation constraint explores whether all entities of one entity set take part in a relationship. If yes this is a total participation, otherwise it is partial. Total participation says that each entity takes part in at least one” relationship and is represented by a bold line (in some books double line).**

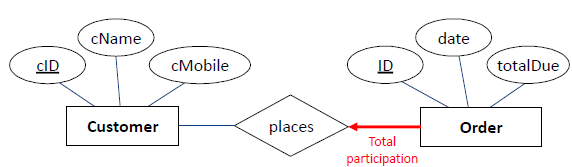
**参与约束探究一个实体集的所有实体是否参与到一个关系中。如果是，这是全部参与，否则是部分参与。总体参与指的是每个实体至少参与一种“关系”，用粗线表示(有些书中用双线表示)。**

* **Example: Every order must belong to a customer, i.e. an order cannot exist on its own without being placed by a customer. The participation of order in places relationship is total participation**

**例如:每个订单都必须属于一个客户，也就是说，没有客户下的订单不能单独存在。秩序在场所关系中的参与是完全参与**

* **A customer may register with the system (becomes a customer) but do not have to place an order. So the participation of customer in places relationship is partial participation**

**客户可以向系统注册(成为客户)，但不必下订单。因此，顾客在异地关系中的参与属于部分参与**

****

* **Strong and Weak entities**
* **An entity is a person, place, object, event or concept.**
* **Entities can be classified into two groups**
* **Strong Entities**
* **"An entity type that exists independently of other entity types"**
* **All previous examples had Strong Entities**
* **Department, Employee**
* **Customer, Order**
* **A strong entity can be identified by its own attributes, meaning a key attribute (a unique identifier) can be chosen from its own attributes**

**一个强实体可以通过它自己的属性来标识，这意味着一个关键属性(一个唯一的标识符)可以从它自己的属性中选择**

* **Weak Entities**
* **The entity cannot exist without the entity with which it has a relationship实体不能在没有与它有关系的实体的情况下存在**
* **The entity has a unique identifier that is partially or totally derived from the parent entity in the relationship**

**该实体具有部分或全部从关系中的父实体派生的唯一标识符**

* **Owner entity set and weak entity set must participate in a one to many relationship set (one owner, many weak entities)**

**所有者实体集和弱实体集必须参与到一对多关系集中**

* **Weak entities are represented by double border rectangle**

**弱实体用双边框矩形表示**

* **Weak entity set must have total participation in this relationship set. Such relationship is called identifying and is represented as “bold”**

**弱实体集必须完全参与这个关系集。这种关系称为识别，并用“粗体”表示。**

* **Example**
* **Buildings at a university contain rooms.**
* **Each building have a building name and building code (e.g. Melbourne Connect-MC, Doug McDonnell-DM)-strong entity**

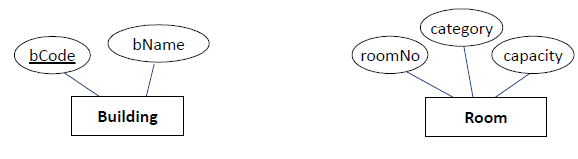
**每个建筑都有一个建筑名称和建筑规范(例如墨尔本Connect-MC, Doug McDonnell-DM)-强大的实体**

* **Each room has room number (e.g. 4.05, 2.01), room category (staff office, lecture theatre, lab) and capacity**

**每个房间都有房间号码(如4.05、2.01)、房间类别(员工办公室、演讲厅、实验室)和容量**

* **Can the attribute roomNo be used as a unique identifier? No, because several buildings have the same room numbers**

**属性roomNo可以用作唯一标识符吗?不，因为好几栋楼都有相同的房间号**

****

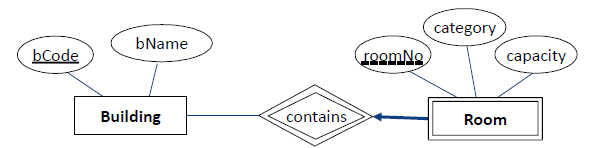
* **Relationship:**
* **One building contains many rooms**
* **A room is contained in one building only**
* **So we need a composite identifier for the room entity, i.e. bCode+roomNo**

**因此，我们需要一个房间实体的复合标识符，即bCode+roomNo**

* **Problem: bCode is not an attribute of room问题:bCode不是room的属性**
* **Solution: Room borrows identifier from building**

**解决方法:房间从建筑物借用标识符**

* **Room is a weak entity**

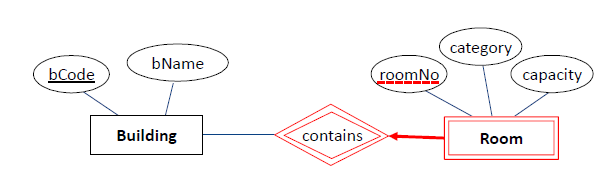
****

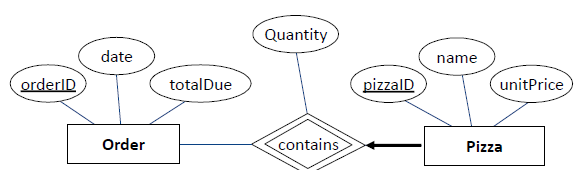
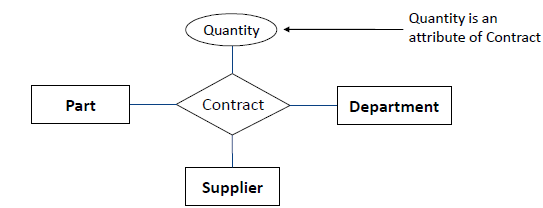
* **Weak entity notation弱实体符号**
* **Entity denoted by double border rectangle用双边框矩形表示的实体**
* **If a weak entity has an attribute that is a partial identifier, it is marked with dashed underline**

**如果弱实体的属性是部分标识符，则用虚线下划线标记**

* **The relationship is shown with double border这种关系用双边表示**
* **Total participation is denoted with a bold arrow**

**总参与率用一个粗体箭头表示**

****

* **Weak entity**
* **An order contains pizza(s)-Many to many**
* **Each pizza can be in many orders-Many to many**
* **Problem: where do we put quantity for each pizza type ordered?**
* ****
* **Ternary Relationships三元关系**
* **In general, we can have n-ary relationships, and relationships can have attributes通常，我们可以有n元的关系，而关系可以有属性**
* ****

1. **Conceptual Design**

* **What are the entities and relationships in the enterprise?**

**企业中的实体和关系是什么?**

* **What information about these entities and relationships should we store in the database?** **我们应该在数据库中存储关于这些实体和关系的哪些信息?**
* **What are the integrity constraints that hold?** **保持的完整性约束是什么?**
* **Conceptual Design Using the ER Model**
* **Design choices:**
* **Should a concept be modelled as an entity or an attribute一个概念应该被建模为一个实体还是一个属性**
* **Should a concept be modelled as an entity or a relationship?**
* **Should we model relationships as binary, ternary, n-ary?**
* **Constraints in the ER Model:**
* **A lot of data semantics can (and should) be captured**

**可以(也应该)捕获许多数据语义**

* **Entity vs Attribute**
* **Should “address” be an attribute of Employees or an entity (related to Employees)?**

**“地址”应该是雇员的属性还是一个实体(与雇员相关)?**

* **Depends upon how we want to use address information, and the semantics of the data:**

**这取决于我们如何使用地址信息，以及数据的语义:**

* **If we have several addresses per employee, address must be an entity**
* **If the structure (city, street, etc.) is important, address should be modeled as an entity**

**如果结构(城市、街道等)很重要，那么地址应该建模为一个实体**

* **Notes on the ER design关于ER设计的注释**
* **ER design is subjective. There are often many ways to model a given**

**ER设计是主观的。通常有很多方法来为一个给定的模型建模**

* **Where to start noun verb analysis名词动词分析从哪里开始**
* **Analyzing alternatives can be tricky, especially for a large enterprise. Common choices include:**

**分析替代方案可能很棘手，尤其是对大型企业而言。常见的选择包括**

* **Entity vs. attribute, entity vs. relationship, binary or n ary relationship.** **实体与属性，实体与关系，二进制或n元关系。**
* **There is no standard notation (we will cover two notations, so far we learned Chen’s notation)**

**没有标准的表示法(我们将介绍两个表示法，到目前为止我们学过Chen的表示法)**

* **Noun-verb analysis solution名词-动词分析解决方案**
* **An investment bank has a number of branches Within each branch a number of departments operate an dare structured in a hierarchical manner. The bank employs a round 3000 staff who are assigned to work in the various departments across the branches**

**投资银行有许多分支机构。在每个分支机构中，有许多部门按等级结构来运作。该银行雇佣了大约3000名员工，他们被分配到各个分支机构的各个部门工作**

* **There are essentially three types of special employees where extra det ails required by the system.** **本质上有三种类型的特殊员工，系统需要额外的检查。**
* **There are dealers who carry out investments who have limits imposed upon them for how much they can spend**

**有些进行投资的交易商对他们能花多少钱有限制**

* **There are IT compliance managers who' s Basel 2 role is required to be stored有些IT合规经理的巴塞尔2角色需要被存储**
* **There are HR managers that need have their assessment number recorded along with other details not specified here**

**有一些人力资源经理需要记录他们的评估编号以及这里没有指定的其他细节**

1. **Attributes**
2. **Attributes have a domain, i.e. the set of possible values for a given attribute**

**属性有一个域，即给定属性的可能值的集合**

1. **Domain is described by the company’s business rules**

**域名由公司的业务规则描述**

1. **Examples:**

* **Attribute Mark at any Australian University must be between 0 and 100 inclusive澳大利亚任何一所大学的属性分数必须在0到100之间**
* **Attribute Grade at the University of Melbourne can only have one of the following values H1, H2A, H2B, H3, P, N, NS**
* **Attributes may share a domain**
* **Phone numbers of staff and customers adhere to the same rules, therefore sharing the same domain**

1. **Required attributes-the attributes that must have a value**

**必需属性——必须有值的属性**

1. **Optional attributes may be left blank可选属性可以留空**
2. **Composite vs Simple attributes**
3. **A composite attribute can be further subdivided into additional attributes**

**复合属性可以进一步细分为其他属性**

* **Address contains street number, street name, city, state, postcode, country**

1. **A simple attribute cannot be subdivided further, e.g., gender, age, year of birth (but not date of birth)**

**一个简单的属性不能再细分，例如，性别，年龄，出生年份(但不是出生日期)**

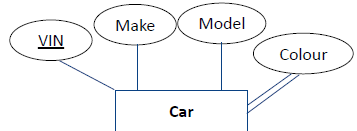
1. **Single valued vs Multi valued attributes**
2. **A single valued attribute can have only a single value, e.g., year of birth, car rego, email address**

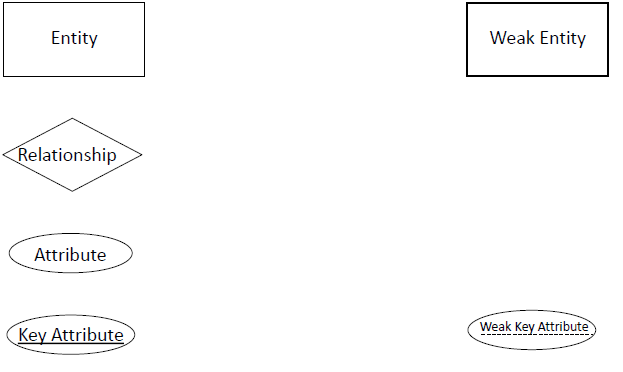
**一个值属性只能有一个值，例如出生年份、汽车rego、电子邮件地址**

1. **A multi-valued attribute can have many values, e.g., a car may have several colours, one for a roof, another for body and another for trim**

**多值属性可以有许多值，例如，一辆汽车可能有几种颜色，一种是车顶，另一种是车身，另一种是装饰**

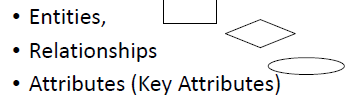
1. **In Chen’s notation a multi-valued attribute is connected to the entity by a double line在Chen的表示法中，多值属性通过双线连接到实体上**

****

1. **Chen notation key shapes**
2. ****

**Logical & Physical Modelling**

1. **So far: Intro to Modelling**
2. **Basic ER modeling concepts**

* ****

1. **Constraints**

* **Key Constraints M:M, 1:M, 1:1**
* **Participation Constraints**
* **Total**
* **Partial**
* **Conceptual Design**

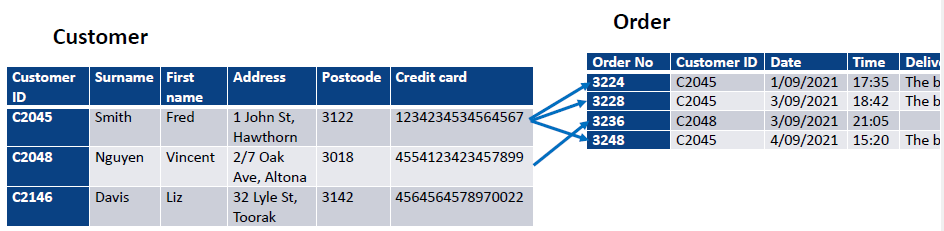
1. **Concepts**
2. **Relational Model关系模型**

* **Relational Data Model关系数据模型**
* **Data Model allows us to translate real world things into structures that a computer can store**

**数据模型允许我们将现实世界的事物转换成计算机可以存储的结构**

* **Many models: Relational, ER, O-O, etc.**
* **Relational Model:**
* **Rows (or Tuples) and Columns (Attributes/fields)**
* **Primary Keys and Foreign Keys to link Relations**

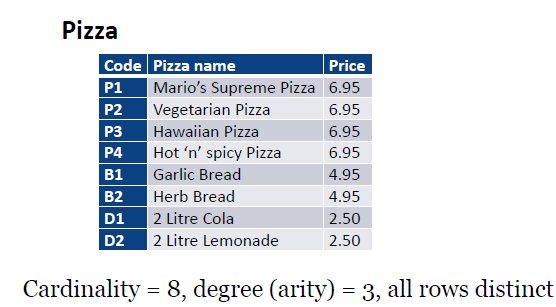
**主键和外键链接关系**

* ****
* **Relational Database: Definitions**
* **Relational database: a set of relations.**
* **Relation: made up of 2 parts:**
* **Schema: specifies name of relation, plus name and type of each column (attribute).**

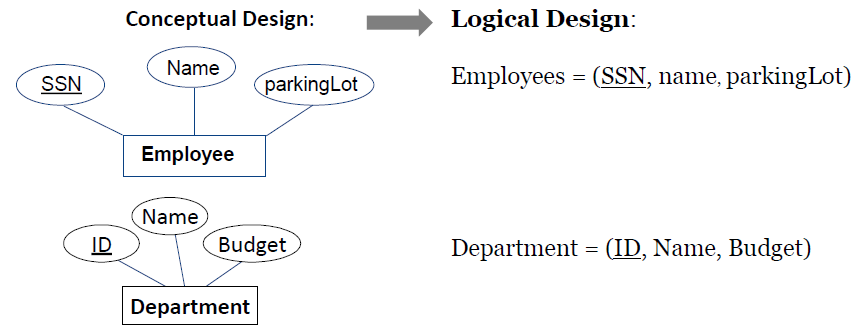
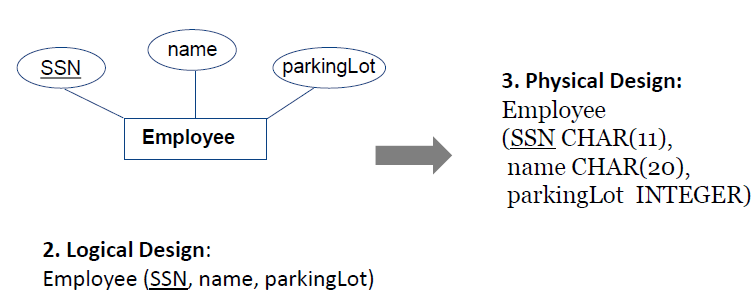
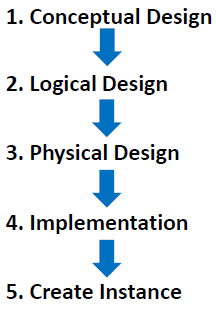
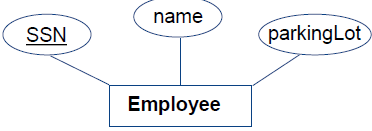
**Schema:指定关系的名称，加上每个列(属性)的名称和类型。**

* **Example:**
* **Pizza (pID : string, pizzaName: string, price : real)**
* **Instance: a table, with rows and columns.** **实例:包含行和列的表。**
* **Number of rows = cardinality基数**
* **Number of columns/fields = degree (or arity)** **列/字段数=度(或秩)**
* **You can think of a relation as a set of rows or tuples.**

**您可以将关系看作一组行或元组。**

* **all rows are distinct所有行都是不同的**
* **no order among rows行之间没有顺序**
* **Example Instance of Pizza Relation**
* ****
* **Logical Design: ER to Relational Model**
* **In logical design entity set becomes a relation. Attributes become attributes of the relation.**

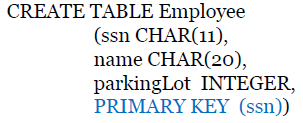
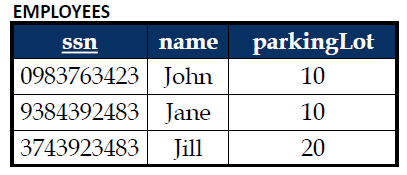
**在逻辑设计中，实体集成为一种关系。属性变成关系的属性。**

* ****
* **ER to Logical to Physical**
* **Conceptual Design:**
* ****
* **The Entire Cycle全生命周期评价**
* ****
* **Conceptual Design**
* ****
* **Logical Design**
* **Employee (SSN, name, lot)**
* **Physical Design:**
* **Employee**

**(ssn CHAR(11),**

**name CHAR(20),**

**parkingLot INTEGER)**

* **Implementation:**
* ****
* **Instance:**
* ****

1. **Keys and Integrity Constraints键和完整性约束**

* **Keys**
* **Keys are a way to associate tuples in different relations**

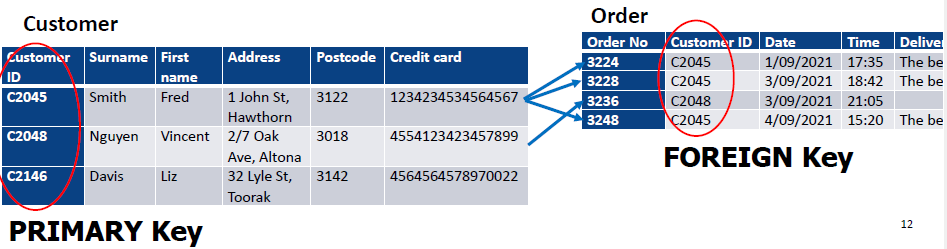
**键是一种以不同关系关联元组的方法**

* **Keys are one form of integrity constraint (IC)**

**键是完整性约束(IC)的一种形式**

* **Example: Only customers can place orders.**
* **Each Foreign Key value in the Order table references a Primary Key value in the Customer table**

**Order表中的每个外键值引用Customer表中的一个主键值**

* ****
* **Primary Keys**
* **A set of fields is a superkey if no two distinct tuples can have same values in all key fields**

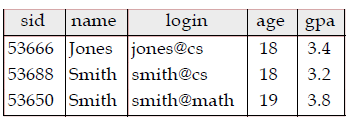
**如果没有两个不同的元组可以在所有关键字段中拥有相同的值，那么一组字段就是一个超键**

* **A set of fields is a key for a relation if it is a superkey and no subset of the fields is a superkey (minimal subset)**

**一组字段是一个关系的键，如果它是一个超键，并且没有字段的子集是超键(最小子集)**

* **Out of all keys one is chosen to be the primary key of the relation. Other keys are called candidate keys.**

**在所有键中，选择一个作为关系的主键。其他键称为候选键。**

* **Each relation has a primary key每个关系都有一个主键**
* ****
* **Selecting the Primary Key**
* **Superkey-a set of fields that contains the key**

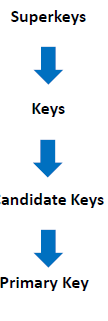
**超级键——一组包含键的字段**

* **Keys are columns that in combination or alone can uniquely identify the tuple (row)** **键是可以组合或单独标识元组(行)的列。**
* **Candidate keys are all the possible key combinations that could be the Primary Key候选键是所有可能成为主键的键组合**
* **Of all candidate keys the database designer identifies the primary key. The primary key is the fewest number of columns that can uniquely identify a key**

**在所有候选键中，数据库设计器标识主键。主键是唯一标识一个键的最少列数**

* **N.B.\* Not all relations will have a key. In cases when there is no key, the database designer will add a surrogate key**

**并不是所有的关系都有钥匙。在没有键的情况下，数据库设计器将添加一个代理键**

* ****
* **Surrogate Keys代理键**
* **A surrogate key is**
* **A key that has no real world / business meaning**
* **Is usually numeric**
* **Is often a sequential number supplied by the RDBMS**
* **Many databases around the world have been created with all their tables using surrogate keys.**

**世界上许多数据库的所有表都是使用代理键创建的。**

* **When modelling business requirements**
* **You speak to clients**
* **You use terms applicable to their business**

**你使用的术语适用于他们的业务**

* **You do not invent terms / fields that do not match their business你不会发明与他们的业务不匹配的术语/领域**
* **You use Natural Keys**
* **Surrogate vs Natural Keys代理键与自然键**
* **If a small college teaches 20 subjects, that business may not have/use subject codes. When identifying a subject, they simply use natural keys such as Subject Name.**

**如果一所小型学院教授20门课程，该企业可能没有/使用学科代码。在标识主题时，它们只使用自然键，如主题名称。**

* **While Subject Name may seem obviously inadequate for a large database, it may be sufficient for Modelling requirements.**

**虽然主题名称似乎明显不适用于大型数据库，但它可能足以满足建模需求。**

* **Forcing a term such as Subject Code into the conversation may confuse clients.**

**在对话中强制使用诸如主题代码之类的术语可能会让客户感到困惑。**

* **Modelling with Natural Keys**
* **Modelling typically uses Natural Keys**
* **Modelling is sometimes the end of the process.**
* **The ER Model that is produced may be the final product**

**所生产的ER模型可能是最终产品**

* **There is no database**
* **Both the client and the modeler learn about the business and its requirements via the modelling process.**

**客户和建模者都通过建模过程了解业务及其需求。**

* **If a Database is required:**
* **Database implementers can choose to add surrogate keys**

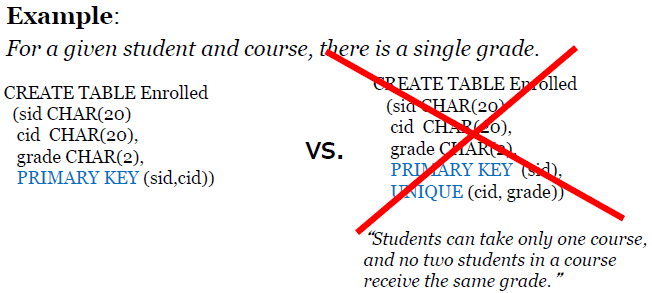
**数据库实现者可以选择添加代理键**

* **When adding a surrogate key you do not lose any data.**

**添加代理键时不会丢失任何数据。**

* **The natural key data is not removed. It's just not the primary key.** **自然的关键数据不会被删除。它不是主键。**
* **Primary and Candidate Keys in SQL**
* **There are possibly many candidate keys (specified using UNIQUE), one of which is chosen as the primary key. Keys must be chosen carefully**

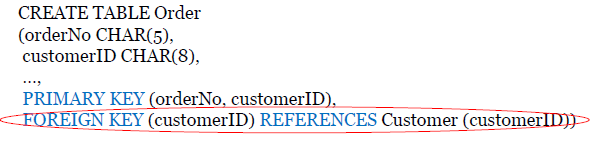
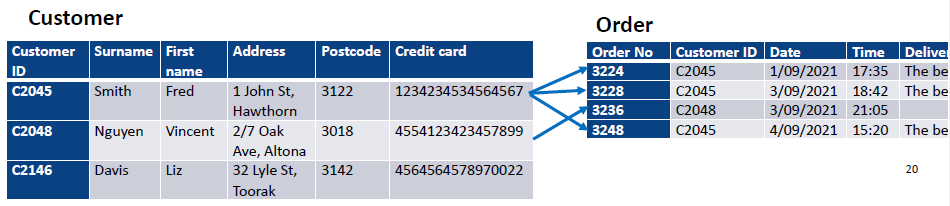
**可能有许多候选键(使用UNIQUE指定)，其中一个被选择为主键。钥匙必须谨慎选择**

* ****
* **Foreign Keys and Referential Integrity外键和引用完整性**
* **Foreign key: A set of fields in one relation that is used to ‘refer to a tuple in another relation. The foreign key must correspond to the primary key of the other relation.**

**外键:一个关系中的一组字段，用于“引用另一个关系中的元组”。外键必须与另一个关系的主键相对应。**

* **If all foreign key constraints are enforced in a DBMS, we say a referential integrity is achieved.**

**如果在DBMS中强制了所有的外键约束，我们就说实现了引用完整性。**

* **Foreign Keys in SQL**
* **Only customers listed in the Customer relation should be allowed to place Orders. customerID is a foreign key referring to Customer table只有在客户关系中列出的客户才可以下订单。customerID是一个引用Customer表的外键**
* ****
* ****
* **Enforcing Referential Integrity实施参照完整性**
* **Consider Customer and Order:**
* **customerNo in Order is a foreign key that references Customer.** **customerNo in Order是一个引用Customer的外键。**
* **What should be done if an Order tuple with a non-existent customerNo is inserted? (Reject it!)**

**如果插入了一个不存在customerNo的Order元组，应该做什么?(拒绝它!)**

* **What should be done if a Customer tuple is deleted?** **如果Customer元组被删除，应该做什么?**
* **Also delete all Order tuples that refer to it?**

**还要删除引用它的所有Order元组?**

* **Disallow deletion of a Customer tuple that is referred to?**

**不允许删除引用的Customer元组?**

* **Set customerNo in Order tuples that refer to it to a default customerNo?**

**设置customerNo in Order元组引用它到一个默认的customerNo?**

* **Integrity Constraints (ICs)** **完整性约束**
* **IC: condition that must be true for any instance of the database; e.g., domain constraints**

**IC:对于数据库的任何实例都必须为真的条件;例如,域约束**

* **ICs are specified when schema is defined.** **ICs在定义模式时指定。**
* **ICs are checked when relations are modified.** **修改关系时检查ICs。**
* **A legal instance of a relation is one that satisfies all specified ICs.**

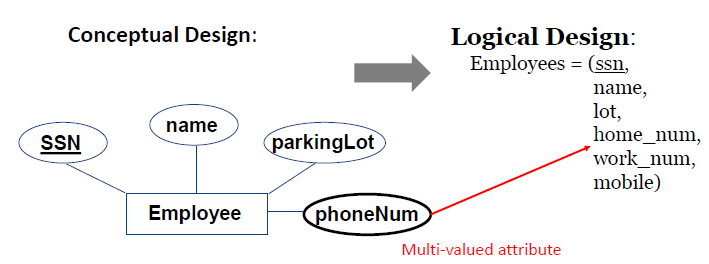
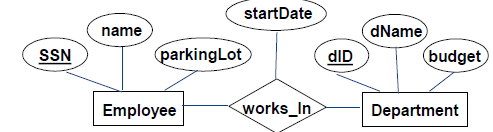
**关系的法律实例是满足所有指定的ICs的实例。**

* **DBMS should not allow illegal instances.** **DBMS不应该允许非法实例。**
* **This is also known as Schema on Write这也被称为写模式**
* **The schema table structure is known in advance of the data to be inserted into it模式表结构在要插入的数据之前就已经知道了**

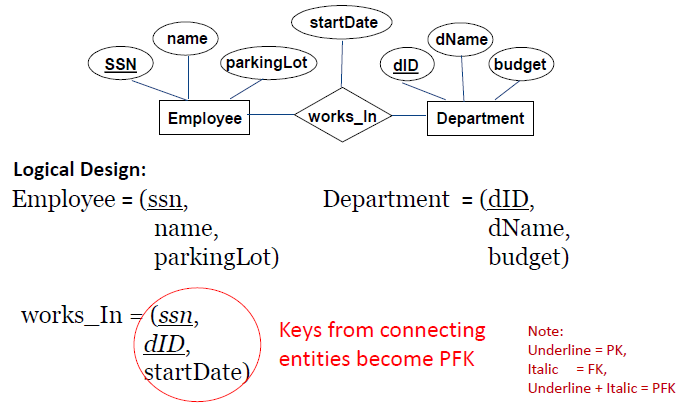
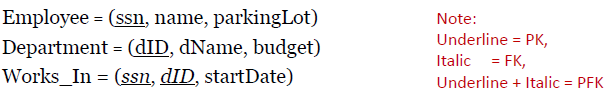
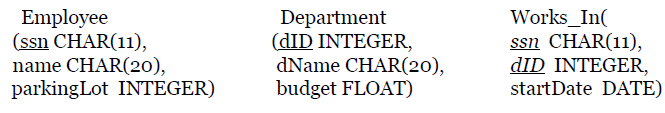
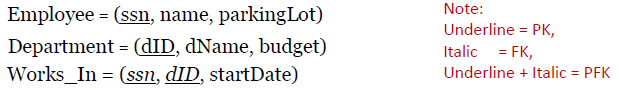
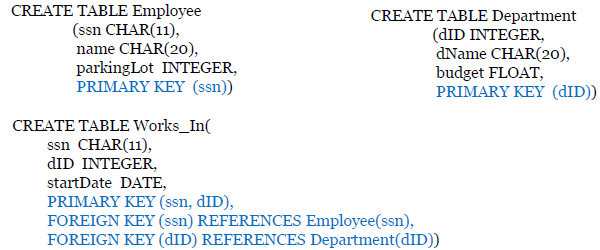
1. **Translating ER to Logical and Physical Model将ER转换为逻辑和物理模型**

* **Multi-valued attributes in logical design逻辑设计中的多值属性**
* **Multi-valued attributes need to be unpacked (flattened) when converting to logical design.**

**在转换为逻辑设计时，需要将多值属性解压缩(平铺)。**

* **For employees we need to capture home phone number, work phone number and mobile对于员工，我们需要获取家庭电话号码、工作电话号码和手机号码**
* ****
* **ER to Logical Design**
* **Conceptual Design:**
* ****
* **Logical Design:**
* **In translating a many to many relationship set to a relation, attributes of a new relation must include:** **在将多对多关系集转换为一个关系时，新关系的属性必须包括:**
* **Keys for each participating entity set (as foreign keys). This set of attributes forms a superkey of the relation.**

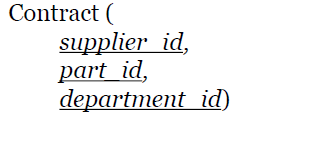
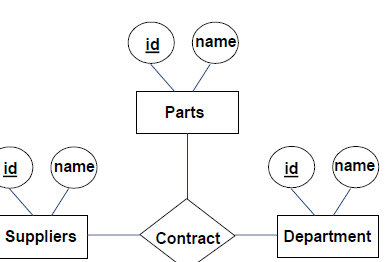
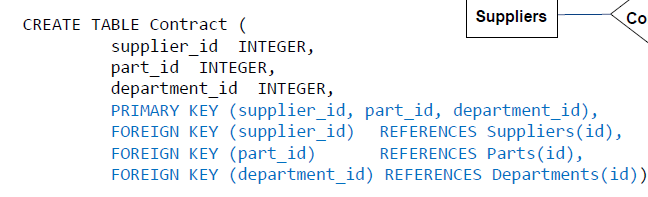
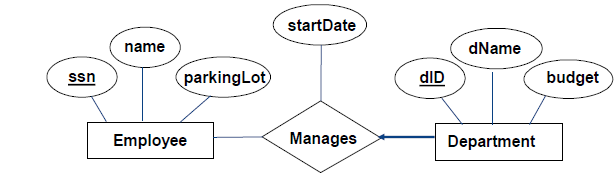
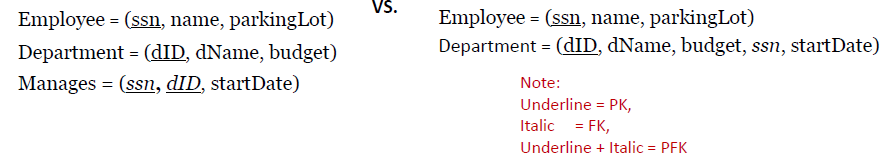
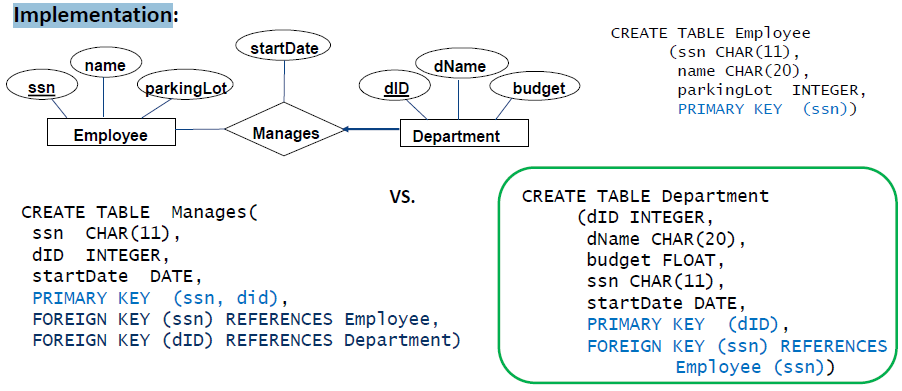
**每个参与实体的键设置(作为外键)。这组属性形成了关系的超键。**

* **All descriptive attributes.**
* ****
* **Logical to Physical Design**
* **Logical Design:**
* ****
* **Physical Design:**
* ****
* **Implementation**
* **Logical Design:**
* ****
* **Implementation:**
* ****
* **ER to Logical Design Example 2**
* **In translating a many to many relationship set to a relation, attributes of the relation must include:**

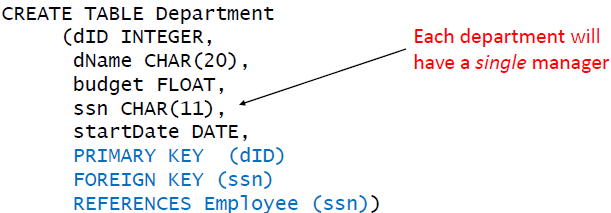
**在将多对多关系集转换为关系时，关系的属性必须包括:**

* **Keys for each participating entity set (as foreign keys). This set of attributes forms a superkey for the relation.**

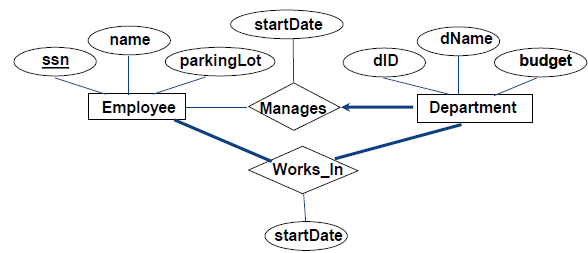
**每个参与实体的键设置(作为外键)。这组属性为关系形成了一个超键。**

* **All descriptive attributes.**
* **Logical Design:**
* ****
* ****
* **Implementation:**
* ****
* **Key Constraints: Logical design**
* **Each department has at most one manager, according to the key constraint on Manages.** **根据管理的关键约束，每个部门最多有一个经理。**
* ****
* **Logical Design:**
* ****
* **Key Constraints in SQL**
* **Implementation:**
* ****
* **Key Constraints rule:**
* **RULE: Primary key from the one side becomes a foreign key on the many side. This is the way to ensure that the key constraint holds**

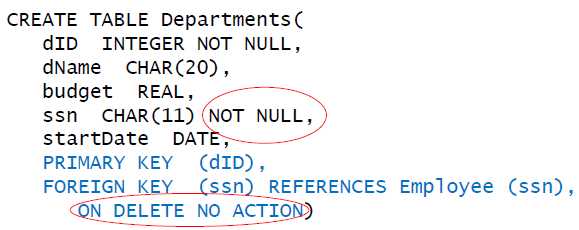
**规则:一边的主键变成了多一边的外键。这是确保关键约束有效的方法**

* ****
* **Review: Participation Constraints**
* **Does every department have a manager?**
* **If so, this is a participation constraint: the participation of Departments in Manages is said to be total (vs. partial)**

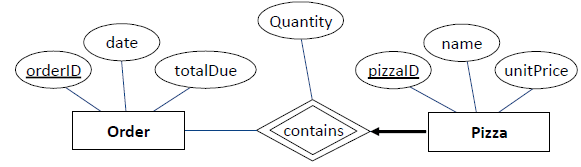
**如果是这样，这就是参与限制:部门参与管理被称为全部参与(而不是部分参与)**

* ****
* **We specify total participation with key words NOT NULL**

**我们使用关键字NOT NULL指定总参与**

* **NOT NULL = this field cannot be empty**
* ****
* **Review: Weak Entities**
* **A weak entity can be identified uniquely only by considering the primary key of another (owner)** **entity**

**弱实体只能通过考虑另一个(所有者)实体的主键来唯一识别**

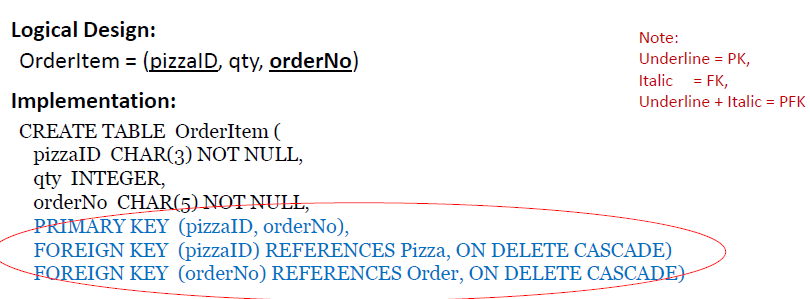
****

* **Translating Weak Entity Sets**
* **Weak entity set and identifying relationship set are translated into a single table.** **弱实体集和识别关系集被转换为单个表。**
* **Contains is not the best entity name when we convert a relationship into an entity**

**当我们将关系转换为实体时，Contains并不是最好的实体名称**

* **When the owner entity is deleted, all owned weak entities must also be deleted.**

**当所有者实体被删除时，所有拥有的弱实体也必须被删除。**

* ****