

資料庫系統筆記 Ch. 4

參考書籍：

1. 《Databases System 7th Edition》— Ramez Elmasri, Shamkant B. Navathe.
2. 《數據庫系統基礎 第六版》— Ramez Elmasri, Shamkant B. Navathe，李翔鷹、劉鎮、邱海艷、陳立軍譯

筆記作者：[黃漢軒](#)

EER 模型

除了本身的 ER 模型之外，還多了：

- 子類別（Subclass）、父類別（Superclass）。
- 特別化（Specialization）、一般化（Generalization）。
- 類別（Category）、並類別（Union type）。
- 屬性（Attribute）、關聯繼承（Relationship Inheritance）的重要機制。

新增了這些東西的 ER 模型，稱為 EER 模型（Enhanced-ER model）。

子類別、父類別與繼承

定義子類別與父類別

- 子類別（Subclass），即把一個實體類型，分成多個有意義的分組或子類別，被分的實體類型就稱為父類別（superclass），例如：
- 對於一間公司來說，EMPLOYEE 可能同時也可以分成 SECRETARY、MANAGER 等。
- 在這邊的 SECRETARY 與 MANAGER 就是子類別。
- 在這邊的 EMPLOYEE 就是父類別。

TODO：補圖

子類別與父類別的一些特性

- 對於子類別與父類別的關聯，就稱為「父類別／子類別關聯」（Superclass/subclass Relationship），同時也可以稱為「父類型／子類型關聯」（Supertype/subtype Relationship），也可簡稱為「類別／子類別關聯」（Class/subclass Relationship）。
- 子類別中的實體必須同時為父類別的實體，例如：秘書（SECRETARY）必須同時也是員工（EMPLOYEE）的實體，秘書的實體不能獨立存在，必須要是以父類別，也就是員工實體的一部分，來分出子類別實體。

TODO：補圖

類型繼承的特性

- 對於從父類別與子類別，他們是從父類別中被分出了各式各樣的子類別。
- 因此子類別應該要有父類別的所有屬性，以及子類別自己本身的一些屬性，如同繼承的概念。
- 若這些子類別除了從父類別繼承的一些屬性之外，也有自己的屬性，那這個子類別也應該被當成一種實體類別。

特殊化與一般化

特殊化

- 對於一個父類別被分成子類別的這個動作，稱為特殊化。
- 例如：將員工（EMPLOYEE）分為秘書（SECRETARY）、工程師（ENGINEER）等等，這個動作稱為特殊化。
- 特殊化後，每個子類別可以擁有自己的屬性，或者擁有自己的關聯類型。
 - 若子類別擁有自己的屬性，這些屬性稱為專有屬性（Specific Attribute）。
 - 若子類別擁有自己的關聯類型，這些關聯類型稱為專有關聯類型（Specific Relationship）。
- 特殊化將父類別與子類別使用圓圈進行連接，圓圈與子類別具有一符號，描述是從哪個父類別被分出了子類別。

TODO：補圖

一般化

- 若今天有兩個類似的實體類型，例如汽車與卡車，我們可以使用一般化，來提取他們的共同屬性。
- 概念上即為特殊化的逆向過程，我們可以利用這樣的方式，來讓兩個屬性使用一個屬性來當作父類別，簡潔 EER Diagram。

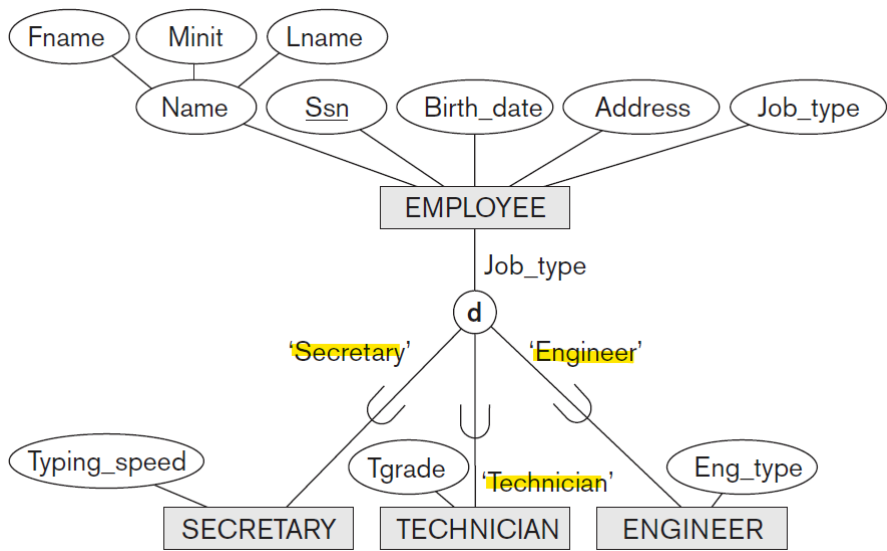
TODO：補圖

特殊化與一般化的層次、約束與特徵

特殊化與一般化的約束

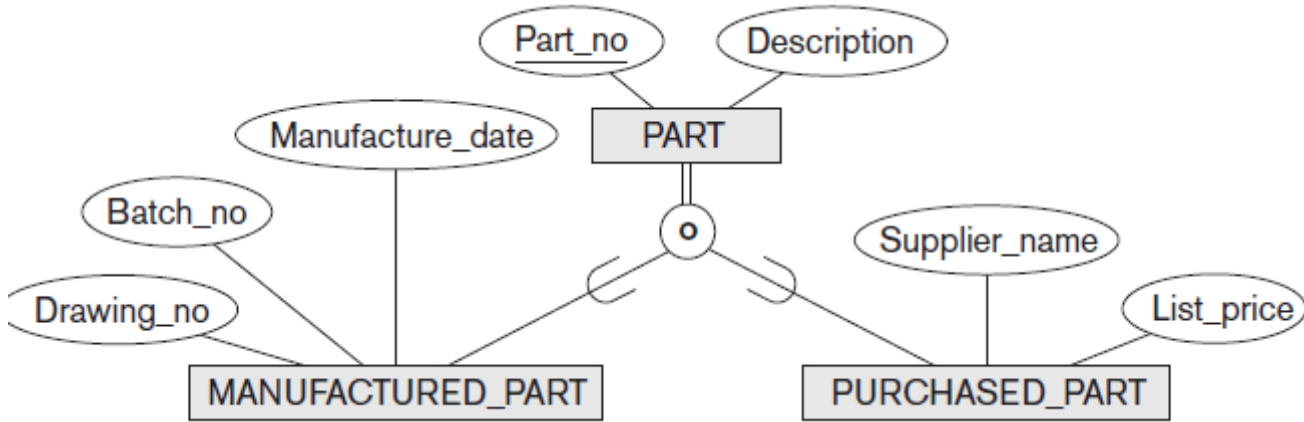
- 對於一個實體類型可以定義多種不同的特殊化
 - 謂詞定義子類
 - 使用謂詞來實作特殊化，例如我們限制職位類別為 Secretary，那麼我們可以用這樣的條件來特殊化子類。
 - 在這邊條件稱為定義謂詞，即為一種約束。
 - 屬性定義特殊化
 - 使用屬性的值來實作特殊化，例如我們使用 Job_type 的值，來分割成三種不同的子類別。
 - 使用者定義特殊化
 - 由用戶單獨一個一個來做特殊化，沒有特定的規則能夠限制他們。

例如下圖使用了屬性定義的特殊化，來實作特殊化的部分。



- 不相交特殊化與重疊特殊化
 - 不相交特殊化（Disjoint），限制特殊化中的種類必須不相交。
 - 必定會特殊化出來兩種以上的子類別。
 - 以圓圈內寫 d 來表示不相交特殊化。
 - 重疊特殊化（Overlap），即為不相交特殊化的相反，其特殊化出來的 member 可能會存在於兩個以上的子類別內。
 - 以圓圈內寫 o 來表示不相交特殊化。

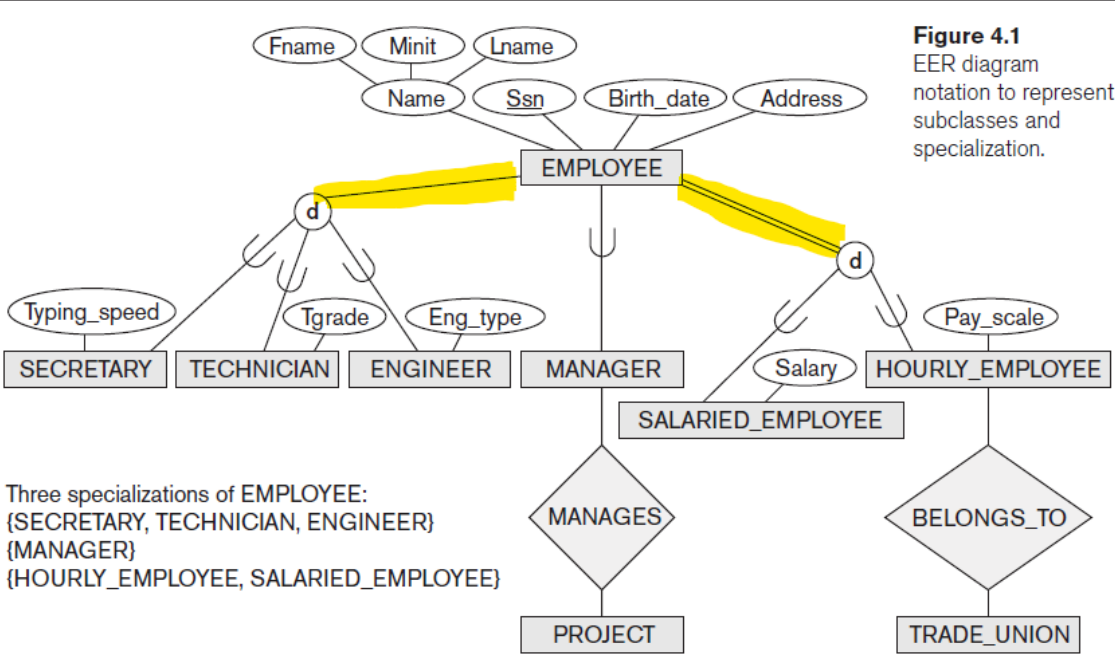
例如下圖描述了重疊特殊化的樣子。



- 完全約束與不完全約束
 - 完全約束：代表一個實體內的每個成員都要參與特殊化後的子類別，以兩條線來表示。
 - 不完全約束，代表一個實體內的成員不一定要參與特殊化後的子類別，以一條線來表示。

例如下圖，將 EMPLOYEE 實體分成兩個子類別使用了兩條線，

代表將所有的 EMPLOYEE 分成了 SALARIED_EMPLOYEE 與 HOURLY_EMPLOYEE。

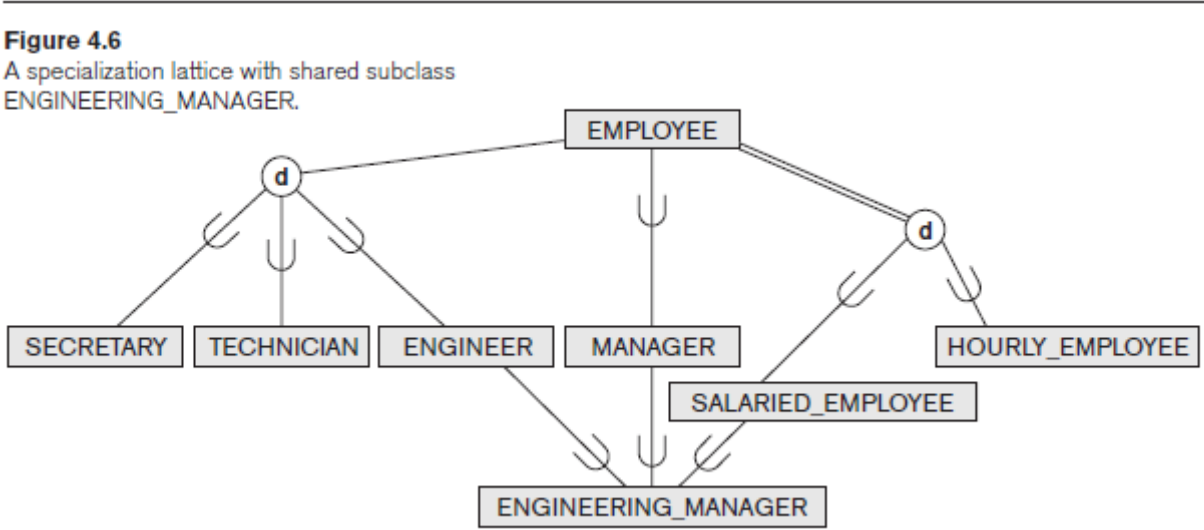


特殊化與一般化的層次與格子

- 對於一個 EER-Diagram 來說：
 - 若規定一個子類別只能擁有一個父類別，這樣就會形成一個樹結構或者有嚴格的層次（Strict Hierarchy）。
 - 這樣每個子類別都只會有單繼承的特性，只會繼承自己唯一的父類別。
 - 若規定一個子類別不一定只能擁有一個父類別，同時是子類別也是父類別的類別，稱為特殊化格子（Specialization Lattice）
 - 特殊化格子擁有多個父類別的繼承，屬於多重繼承，這個子類稱為共享子類。

如下圖，若成為工程師經理需要同時是經理與工程師，

那麼就會將 EER-Diagram 畫成以下形式，其中 ENGINEERING_MANAGER 即為特殊化格子，也是共享子類。



同樣的概念也適用於一般化。

聯集型態

- 當我們同時想要聯集兩個以上的類型，成為唯一一種類型時，可以使用聯集型態，使用圓圈中間畫 U 來表示。
- 這樣可以更方便讓我們來規劃 EER-Diagram，將多個類型聯集成一種類型，再接著做處理。
- 需要注意的是：
 - 與一般化的地方不同的是，一般化主要是提起共同屬性來創立父屬性
 - 聯集型態是將多個屬性聯集在一起，並不是提取共同。

如下圖，我們可以將 CAR 與 TRUCK 聯集成 REGISTERED_VEHICLE 的類型，使用圓圈中間畫 U 來表示。

