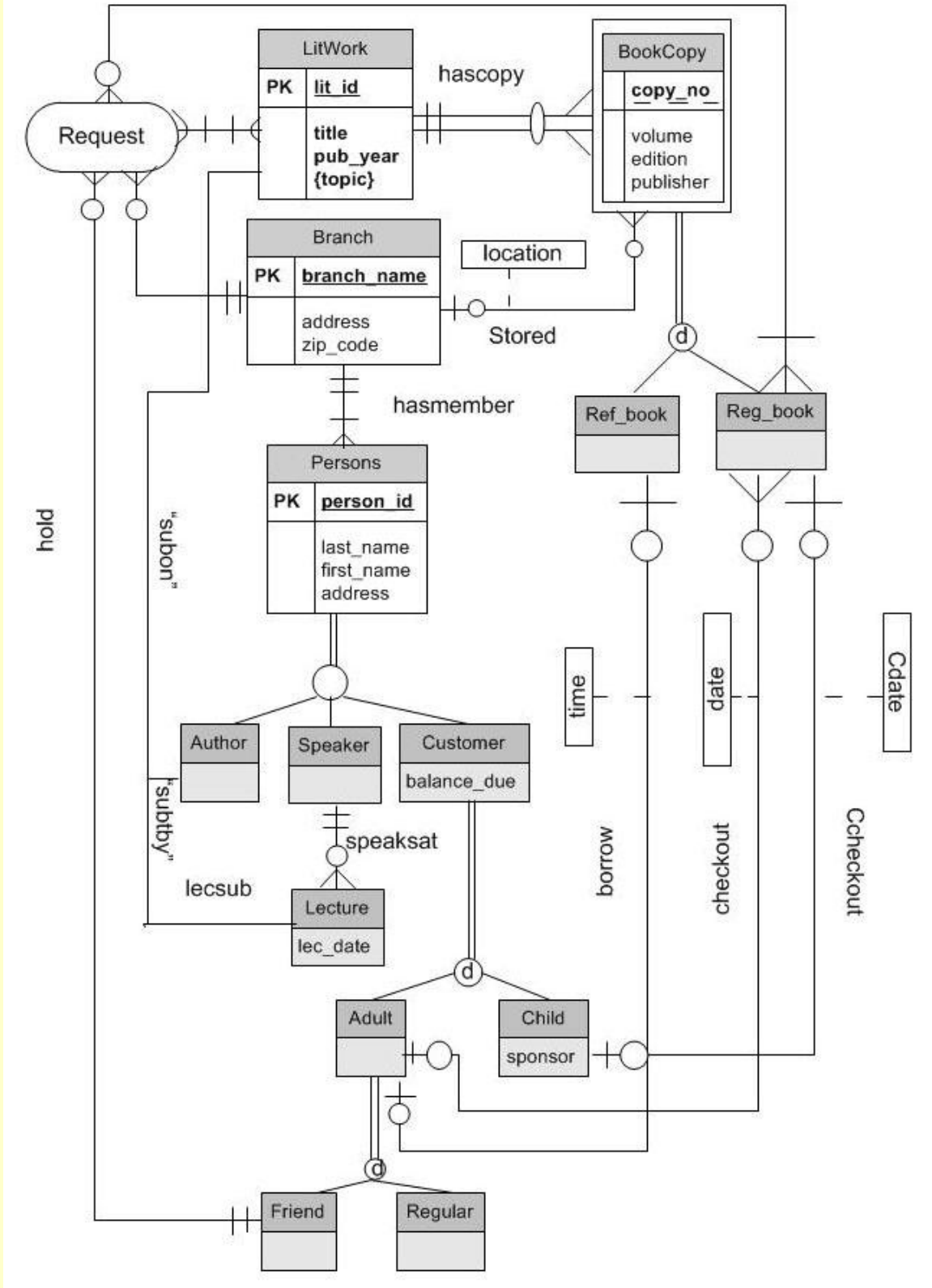


# **Library Database**

## **Entity / Relationship Diagram**

Curt Ireton

# Library Database – E/R Diagram



## Explanation of Library Database Entity-Relationship Diagram

### Entities

The **LitWork** entity has a **lit\_id** as a primary key, the attributes are **title**, **pub\_year**, and multi-valued attribute **topic**, because each **LitWork** may have several **topics**.

The **BookCopy** entity is a weak entity with a partial key of **copy\_no**. It is a weak entity because several instance of **BookCopy** may have the same **copy\_no**. It is identified by **hascopy** because we need not only the **copy\_no**, but also the defining **LitWork** and **lit\_id**. **BookCopy** is the parent in a total generalization hierarchy to its children – disjoint entities **Ref\_book** and **Reg\_book** (which are Reference book and Regular book). The hierarchy is total because all **BookCopys** must be either reference or regular, it is disjoint because a **BookCopy** cannot at the same time be both.

**Request** is an associative entity. It tells what **Friend** requested a **Reg\_book** to be held, what the **LitWork** is, what **Branch** they would like to pick it up in, and **Requests** a copy of the **Reg\_book**.

The entity **Branch** has a primary key **branch\_name** and attributes **address** and **zip\_code**.

The entity **Persons** has a primary key of **person\_id** and attributes of **last\_name**, **first\_name**, and **address**. It is the parent in a generalization hierarchy to children entities **Author**, **Speaker**, and **Customer**. It is total because all persons in the database are one of these. It is overlapping because one could also be another (for instance a **Speaker** may also be a **Customer**). The

**Customer** entity has an attribute also of **balance\_due**. **Customer** is a parent to children entities **Adult** and **Child** in a total hierarchy because all **Customers** can only be **Adults** or **Children**, it is disjoint because they cannot be the same. The **Child** entity has an attribute of **sponsor**. The **Adult** entity is a parent to child entities **Friend** and **Regular** in a total hierarchy because all **Adult Customers** must be either **Friends** or **Regular**. It is disjoint because **Friends** are not **Regular Customers** (nor vice versa).

The entity **Lecture** has an attribute of **lec\_date**.

#### Relationships

The relationship **hascopy** is an identifying relationship between **LitWork** and the weak entity **BookCopy** is defines **BookCopy** as having a particular **lit\_id**. It is a **1-M** relationship because one **LitWork** may have several **BookCopys**, but a **BookCopy** may belong to only one **LitWork**. The participation for **Litwork** is partial, since a **LitWork** may or may not have a **BookCopy**, but total for **BookCopy** since **BookCopy** must belong to a **LitWork**.

The relationship **stored** has an attribute of **location** to tell if the book is checked-out, borrowed, in transit or at a branch. This attribute is needed because the **location** is updated when a book is returned, arrives at a branch, or is located somewhere other than its regular branch.

#### The relationships between the associative entity **Request** and other entities:

The relationship **hold** connects **Request** to **Friend** to show what **Friend** has **Requested** the book. It is a **1-M** relationship because one **Friend** may request several books, but each **Request** is made by only one **Friend**. A

**Friend** may or may not **Request** books so the participation constraint is partial, but a **Request** must always be made by a **Friend**, so the constraint is total.

Between **Request** and **Reg\_book** there is a **M-M** relationship since there may be many **Requests** for many **Reg\_books**, A **Request** must always be for a **Reg\_book** so the constraint is total, but a **Reg\_book** may or may not be **Requested** so the constraint is partial.

There is a **M-M** relationship between **Request** and **LitWork** since many **LitWorks** may have many **Requests**. The **Request** constraint is total since the **Request** must be about a **LitWork**, and the **LitWork** constraint is total because only **LitWorks** can be **Requested**.

There is a **1-M** relationship between **Request** and **Branch** since the book is held at one **Branch** (which may be **Requested** by **Friend**), and many **Reg\_books** may be checked out. A **Request** will always be held in **Branch**, but a **Branch** may or may not have a **Request**.

#### The Ternary Relationship

There is a ternary relationship between the entities **LitWork**, **Author**, and **Lecture**. This information connects the three entities to define the **Lecture** that is being given.

The relationship “**subon**” is what **Litwork** the subject pertains to, “**subby**” pertains to the **Author** that wrote the work, and **lebsub** is the relationship that defines the **Lecture** subject.

Note: This relationship connects the entity **Author**, with its inherited attributes from **Person**, to **Litwork**

### Check-out Relationships

The relationship **borrow** has an attribute of **time** to tell what time the **Ref\_book** is **borrowed**. This will also be calculated in the database to tell when the **Ref\_book** is due back and then deleted when it is returned. It is a **1-1** relationship because only one **Ref\_book** may be **borrowed** at a time, and only one **Adult** will **borrow** it. There is a partial constraint on both sides of this relationship since a **Ref\_book** may or may not be **borrowed** and an **Adult** may or may not **borrow** a **Ref\_book**.

The relationship **checkout** has an attribute of **date** which will update the database what the due date is (via calculation of **checkout date** plus, say, two weeks) and when the **Reg\_book** is returned. It is a **1-M** relationship because one **Adult** may **checkout** as many **Reg\_books** as they want, but **Reg\_books** can be only **checkedout** by one **Adult** at a time. There is a partial participation constraint on both sides since a **Reg\_book** because a **Reg\_book** may or may not be **checkedout** and an **Adult** may or may not **checkout** a **Reg\_book**.

The **Ccheckout** (for **Child** checkout) relationship has an attribute of **Cdate** to indicate the date that a **Child** **checkedout** a **Reg\_book**. This will update the database what the due date is and when the **Reg\_book** is returned. There is a **1-1** cardinality because only one **Child** can checkout only one **Reg\_book** at a time. There is partial participation on both sides because a **Child** may or may not **(C)checkout** a **Reg\_book**, and a **Reg\_book** may or may not be **(C)checkedout** to a **Child**.

Note: Since these entities have inherited their attributes from the entity **Persons**, the information passed at **checkout**, **Ccheckout**, and **borrow** will include the **Customer's** information, and the **Branch** they belong to via the relationship **hasmember**.

#### Other Relationships

The relationship **hasmember** connects **Persons** with **Branches**. It is a **1-M** relationship because one **Branch** has many members (**Persons**), and one **Person** can be a member of only one **Branch**. There is total participation constrain on both sides because every **Branch** has at least one member (**Person**) and each **Person** is a member of only one **Branch**.

The relationship **speaksat** is a **1-M** relationship because one **Speaker** may give many **Lectures**, but a **Lecture** can only be given by one **Speaker**. The constraint is partial for a **Speaker** because they may or may not give **Lectures**, but total for **Lecture**, because it must be given by a **Speaker**.