Entity Relationship Diagram (ERD):
Getting Your ERD Right

by Ts. Abdul Razak Hussain, Dept. of Software Engineering

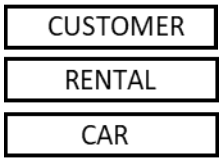
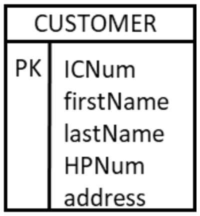
1. ERD? DFD? Which one?

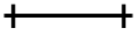
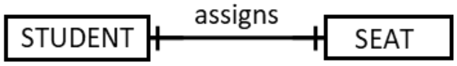
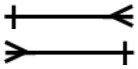
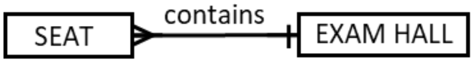
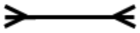
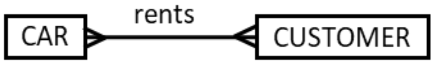
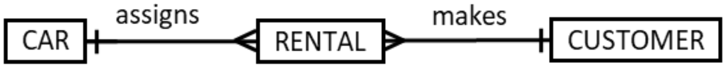
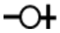
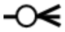
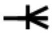
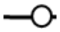
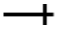
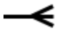
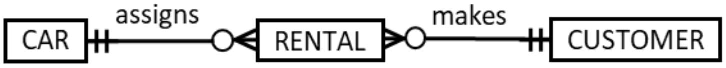
Entity relationship diagram (ERD) and data flow diagram (DFD) are two common diagrams used to graphically represent elements at certain stages in a system development lifecycle. While DFDs are used in the analysis stage to represent the flow of data, ERDs are used in the database design stage to conceptually illustrate the data elements and the interactions/relationships among the entities. These two diagrams are meant to be used in conjunction with one another; neither of them can fully represent the entire application.

Over the years, students may have overlooked the importance of ERD in illustrating WHICH data elements (attributes) should be grouped together as entities and HOW these entities are related with other entities. While a comprehensive ERD is able to convey data-driven business rules, it is unable to capture process-driven constraints. This paper highlights the essential elements of an ERD and three common mistakes when creating ERDs.

2. ERD Essential Elements

There are at least three major ERD notations: 1. Chen notation, 2. Crow's Foot, and 3. UML Class Diagram notation. Further readings can be found in the references section. This article briefly discusses the Crow's Foot notation. The three main elements in an ERD are entities, attributes, and relationships.

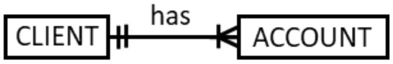
ERD Element	Description
Entity 	+ shown as a rectangle, an entity represents either: i. a **tangible** object (person, object, place) – CUSTOMER, PATIENT, CAR, BRANCH ii. an **intangible** object (event, concept) – RENTAL, REGISTRATION, ACCOUNT + entity name is usually a singular noun; an entity instance is the occurrence of a particular entity. + a higher level ERD may only display entity names without attributes; a comprehensive ERD should illustrates both attributes and relationships.
Attributes  CUSTOMER PK ICNum firstName lastName HPNum address	+ attributes are descriptive elements (or fields) of an entity: i. **simple/atomic** (e.g. street, city, postcode, state) *vs.* **composite** (address) *[a composite attribute can be split into simple attributes]* ii. **single-valued** (ICNum) *vs.* **multi-valued** (HPNum) [*a customer have only one ICNum but may have many HP numbers*] iii. **stored** (dateOfBirth) *vs.* **derived** (age) [*age = currentDate – dateOfBirth*] + certain attributes can be classified as **primary keys** (PK) and/or **foreign keys** (FK) + both PK and FK may consist of a single or a combination of attributes + a good ERD should list all relevant attributes which can be used for future queries. For example, if we are interested in knowing the percentage of our male and female customers and their corresponding age groups, then we must include the attributes gender and dateOfBirth/ageGroup in entity Customer.
Relationship	+ relationships between entities are dictated by business rules or accepted organizational policies; it is shown as a line between entities. All example shown

	in this article are of **binary** relationship (involving two entities); other typical types are unary, ternary and quaternary relationship types.
Connectivity: (one-to-one) 1:1 relationship 	+ relationships can be described as connectivity or cardinality: **connectivity** is the way entities are related or connected:  EACH student is assigned to ONE seat; EACH seat is assigned to ONE student. (Rarely found in actual implementation)
(one-to-many) 1:M relationship 	 EACH exam hall contains MANY seats; EACH seat is contained in ONE exam hall.
(many-to-many) M:N relationship 	 EACH customer rents MANY car (over a period of time); EACH car is rented by MANY customers. As a rule of thumb, M:N relationships MUST be converted to two 1:M relationships for proper relational database implementation. Entity Rental is a **bridge** (or **composite**) entity. 
Cardinality:  zero or one  zero or many one and only one  one or many	**cardinality** indicates the minimum and maximum participation between entity instances occurrences. It makes use of three basic symbols:  = optional or zero  = one  = many  EACH Car is assigned to zero or may Rental; EACH Rental refers to one and only one Car. EACH Customer makes ZERO or MANY Rental; EACH Rental is made by ONE and ONLY ONE Customer.

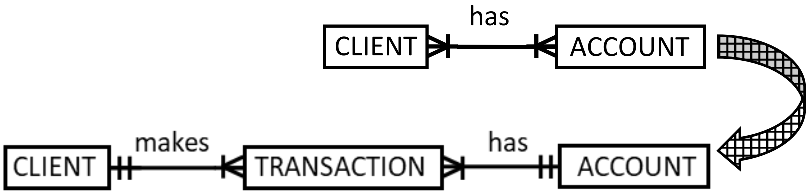
3. Three ERD Common Mistakes to Avoid

3.1 Incorrect connectivity and cardinality between entities.

This may be the result of poorly defined business rules or misinterpreted business rules. In a banking scenario for example, we tend to come up with this ERD - EACH client has ONE or MANY accounts; EACH account belongs to ONE and only ONE client.



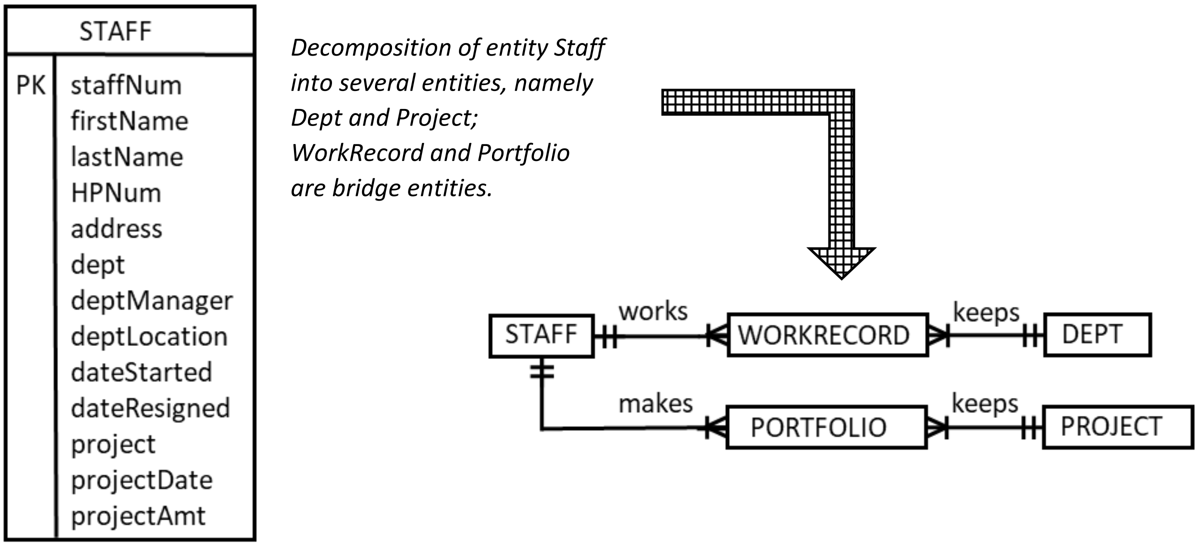
In reality, there are joint accounts that could be opened by more than one clients (many-to-many relationship), one client may have more than one joint accounts. A bridge entity need to be introduced whenever there is a M:N relationship; the transaction entity is used to keep track of the account withdrawals and deposits by the clients.



introduction of a bridge entity Transaction

3.2 Grouping of irrelevant attributes within entities

Given an entity Staff, one may have the tendency to include attributes that may or may not be uniquely or directly related to Staff. Entity Staff need to be decomposed into entity Dept (department) and Project to ensure proper record keeping – reduced data redundancy.

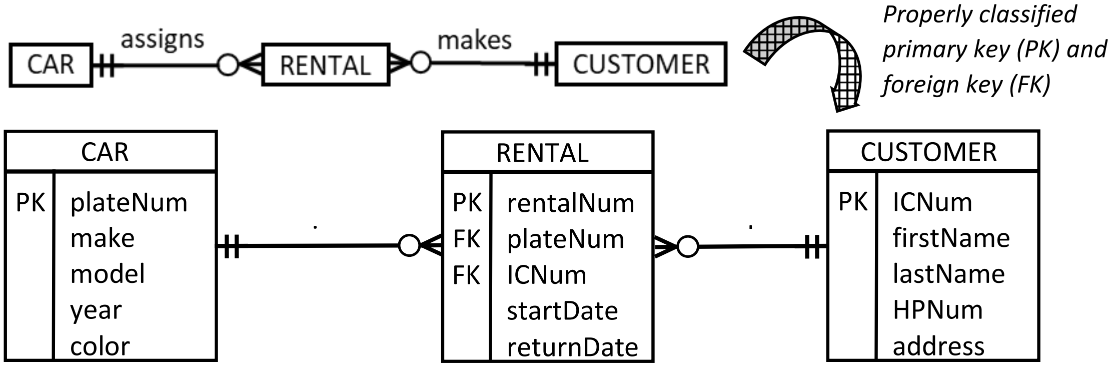


STAFF	
PK	staffNum firstName lastName HPNum address dept deptManager deptLocation dateStarted dateResigned project projectDate projectAmt

Decomposition of entity Staff into several entities, namely Dept and Project; WorkRecord and Portfolio are bridge entities.

3.3 Lack of foreign key classification in bridge entities

Whenever a bridge entity is introduced as a result of the decomposition of a binary many-to-many relationship, its primary and foreign key attributes must be clearly classified. Failure to do so may results in improper database implementation that may violate referential integrity. In the following example, (in entity Rental) attributes plateNum and ICNum are classified as foreign keys referring to entity Car and Customer, respectively.



Properly classified primary key (PK) and foreign key (FK)

CAR	
PK	plateNum make model year color

RENTAL	
PK FK FK	rentalNum plateNum ICNum startDate returnDate

CUSTOMER	
PK	ICNum firstName lastName HPNum address

4. In A Nutshell

ERD graphically illustrate the entities, attributes, and the relationship between the entities. Used primarily in the database design stage, it greatly facilitates both communication and verification of business rules. Current ERD drawing tools may include, among other, data dictionary capabilities that make database implementation seems effortless. In spite of the availability of these complex tools, solid and practical understanding of ERD should not be compromised.

5. Resources

ER Diagram: Entity Relationship Diagram Model | DBMS Example
https://www.guru99.com/er-diagram-tutorial-dbms.html

Entity Relationship Diagram (ERD) Tutorial - Part 1
https://www.youtube.com/watch?v=QpdhBUYk7Kk

Crow's Foot Notation
https://www.lucidchart.com/pages/er-diagrams (last accessed: 25 September 2021)
https://www.vivekmchawla.com/erd-crows-foot-relationship-symbols-cheat-sheet/ (last accessed: 15 September 2021)
http://jackzheng.net/teaching/archive/cis3730-2010-fall/files/1.5-erd.pdf (last accessed: 30 September 2021)
https://www.vertabelo.com/blog/why-need-an-er-diagram/ (last accessed: 10 September 2021)
https://dataedo.com/kb/tools/datagrip/create-database-diagram (last accessed: 30 September 2021)

UML Class Diagram Notation
https://www.conceptdraw.com/examples/database-symbol-in-uml (last accessed: 15 September 2021)
https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/ (last accessed: 15 September 2021)
https://creately.com/diagram-type/objects/class-diagram (last accessed: 30 September 2021)

Chen Notation
https://www.vertabelo.com/blog/chen-erd-notation/ (last accessed: 10 September 2021)
https://creately.com/diagram/example/io05krr81/chen-notation-erd-%5Bclassic%5D (last accessed: 10 September 2021)
https://www.conceptdraw.com/examples/what-is-chen-notation (last accessed: 10 September 2021)

"Thank you, Sir. And would you like to be added to my database?"

Reprinted from Funny Times / PO Box 18530 / Cleveland Hts. OH 44118
phone: 216.371.8600 / email: ft@funnytimes.com