RELATIONAL DATABASES

BSc IoT

Objectives

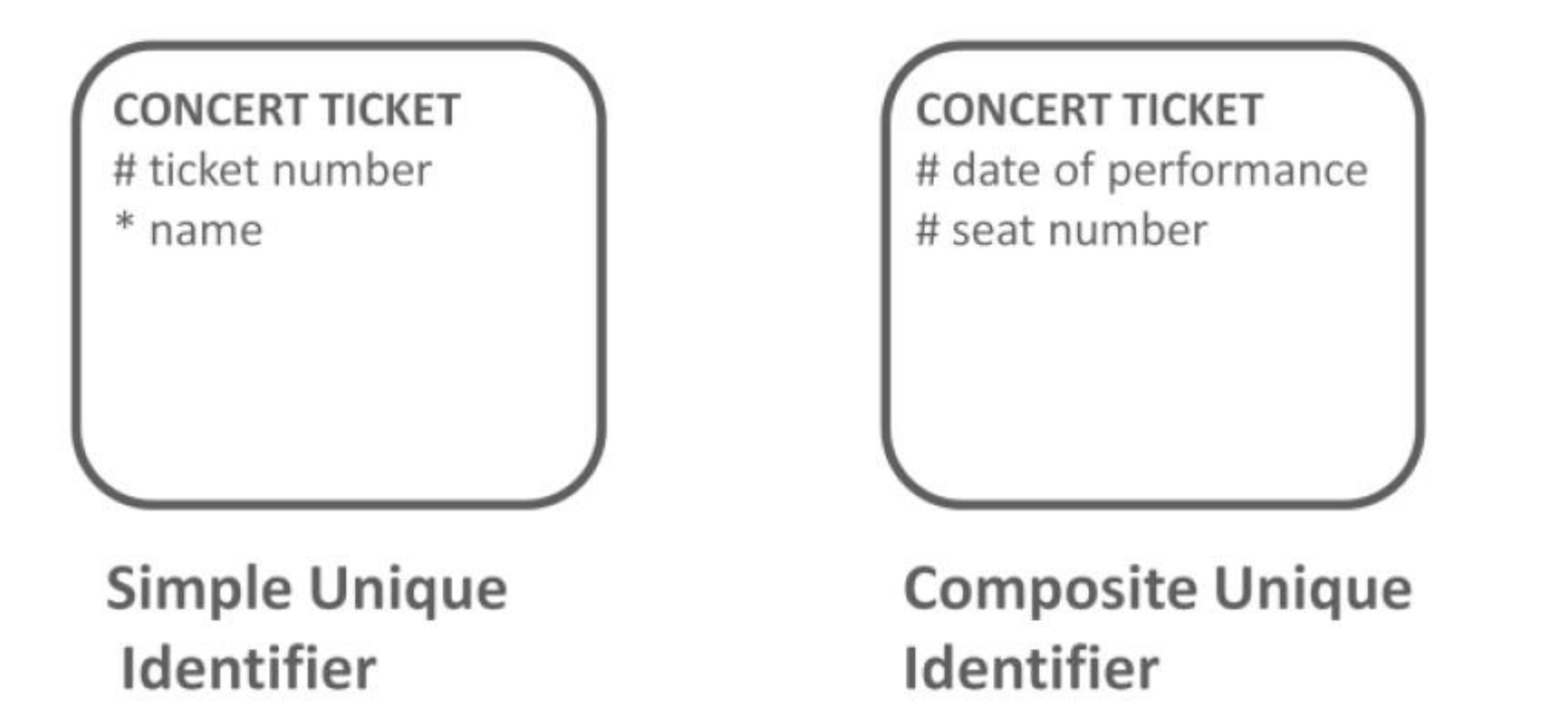
- Define different types of unique identifiers (UIDs)
- Define a candidate UID and explain why an entity can sometimes have more than one candidate UID
- Analyse business rules and choose the most suitable primary UID from the candidates
- Recognise and discuss the issues of identification in the real world

Purpose

- The unique identifier (UID) is very important in relational databases.
- It is the value or combination of values that enables the user to find that one unique item among all the rest
- Identifying just the right attribute, or combination of attributes and relationships, is a skill that any database designer must master
- The unique identifier enables you to find your record in a file, a particular card in a deck of cards, your package in a warehouse, or a specific piece of data in a database.

Simple UIDs vs. Composite UIDs

- A UID that is a single attribute is a simple UID
- Sometimes a single attribute is not enough to uniquely identify an instance of an entity
- If the UID is a combination of attributes, it is called a composite UID



CONCERT TICKET
ticket number
* name

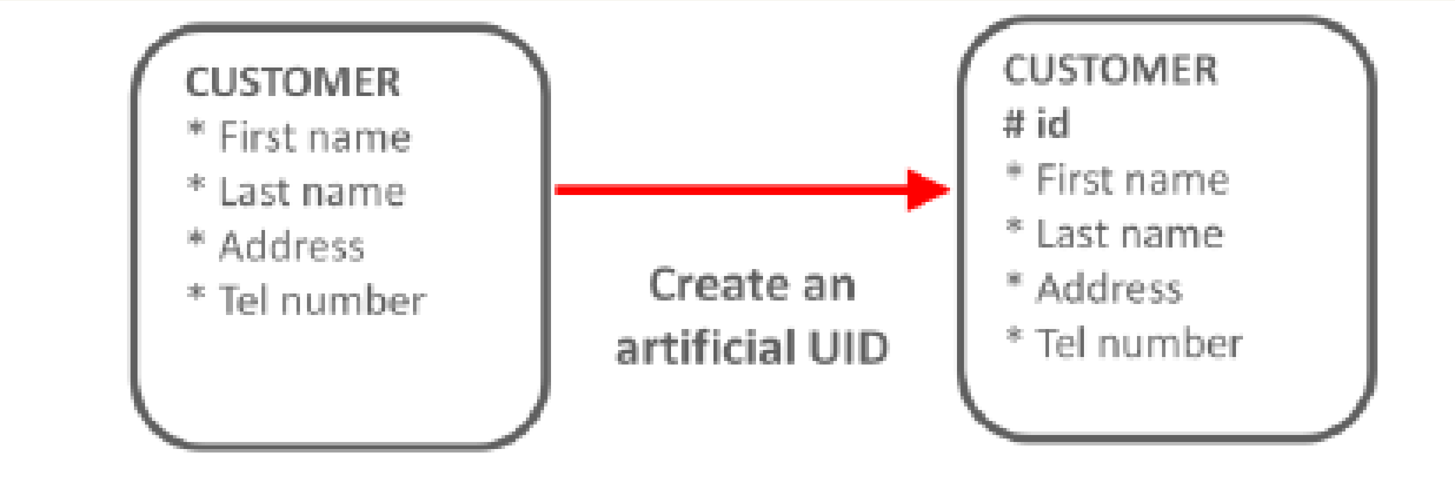
Simple Unique Identifier

CONCERT TICKET
date of performance
seat number

Composite Unique Identifier

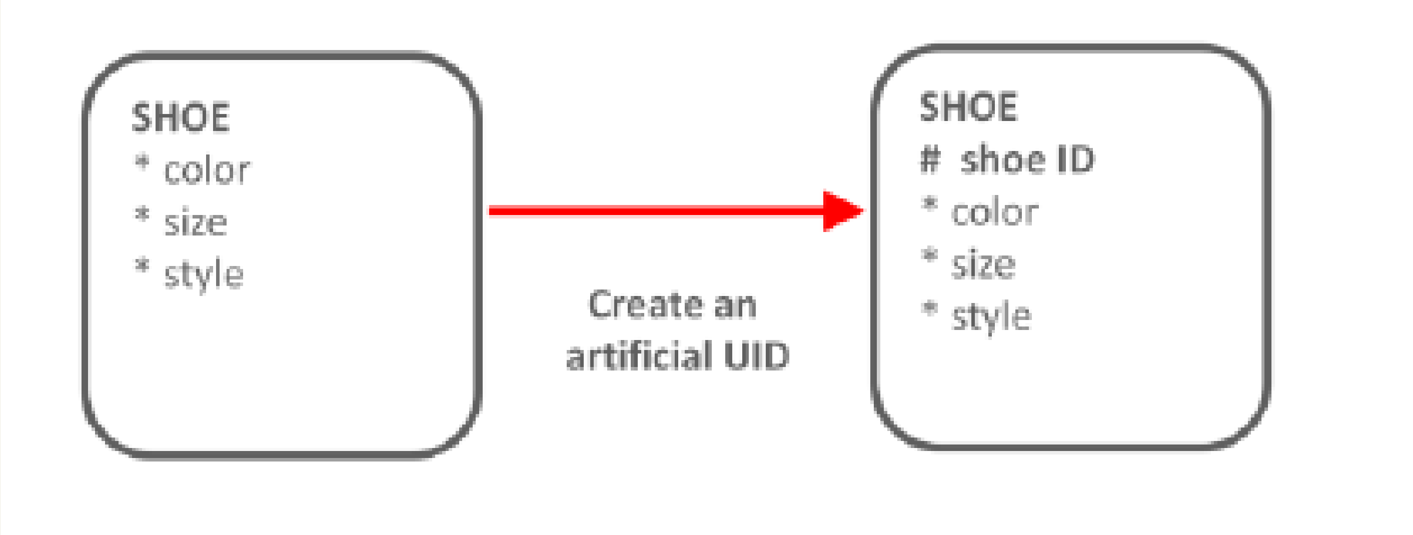
Artificial UIDs

- Artificial UIDs are those that don’t occur in the natural world but are created for purposes of identification in a system
- People are not born with numbers, but a lot of systems assign unique number to identify people: student numbers, customer ids, etc.



Artificial UIDs

- A shoe has a colour, a size, a style, but no truly descriptive number
- However a shoe store will assign unique numbers to each pair of shoes so they can be uniquely identified

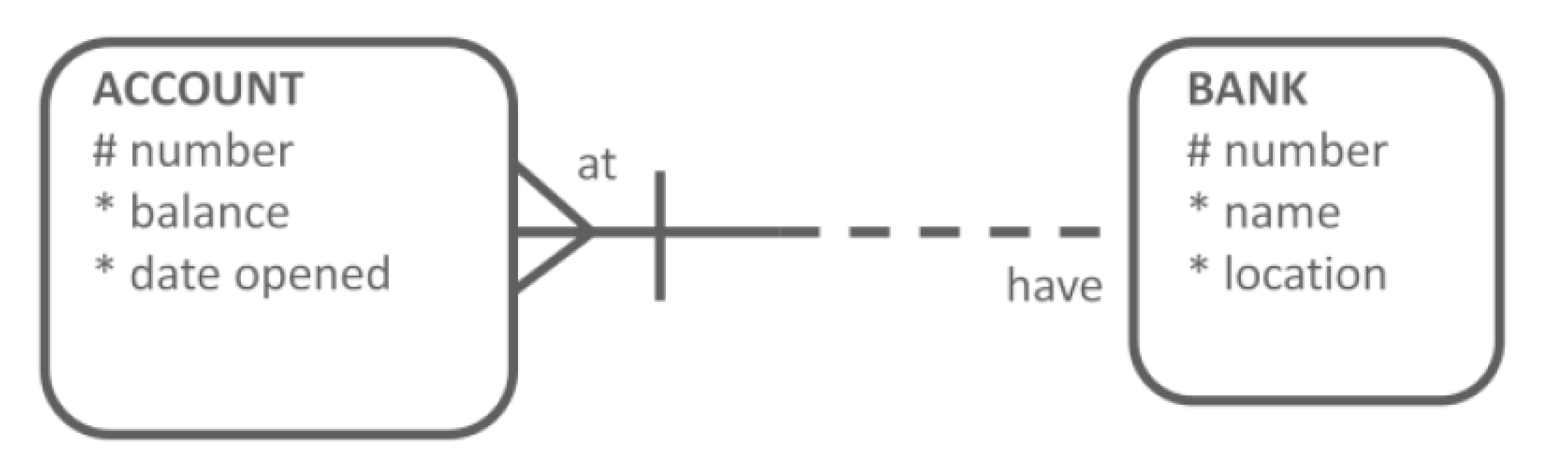


Artificial UID

- It is often simpler to create an artificial attribute and make it the unique identifier.
- Sometimes it is more appropriate for the business to use an artificial number, then when customers make contact they use their customer number rather than name address etc.

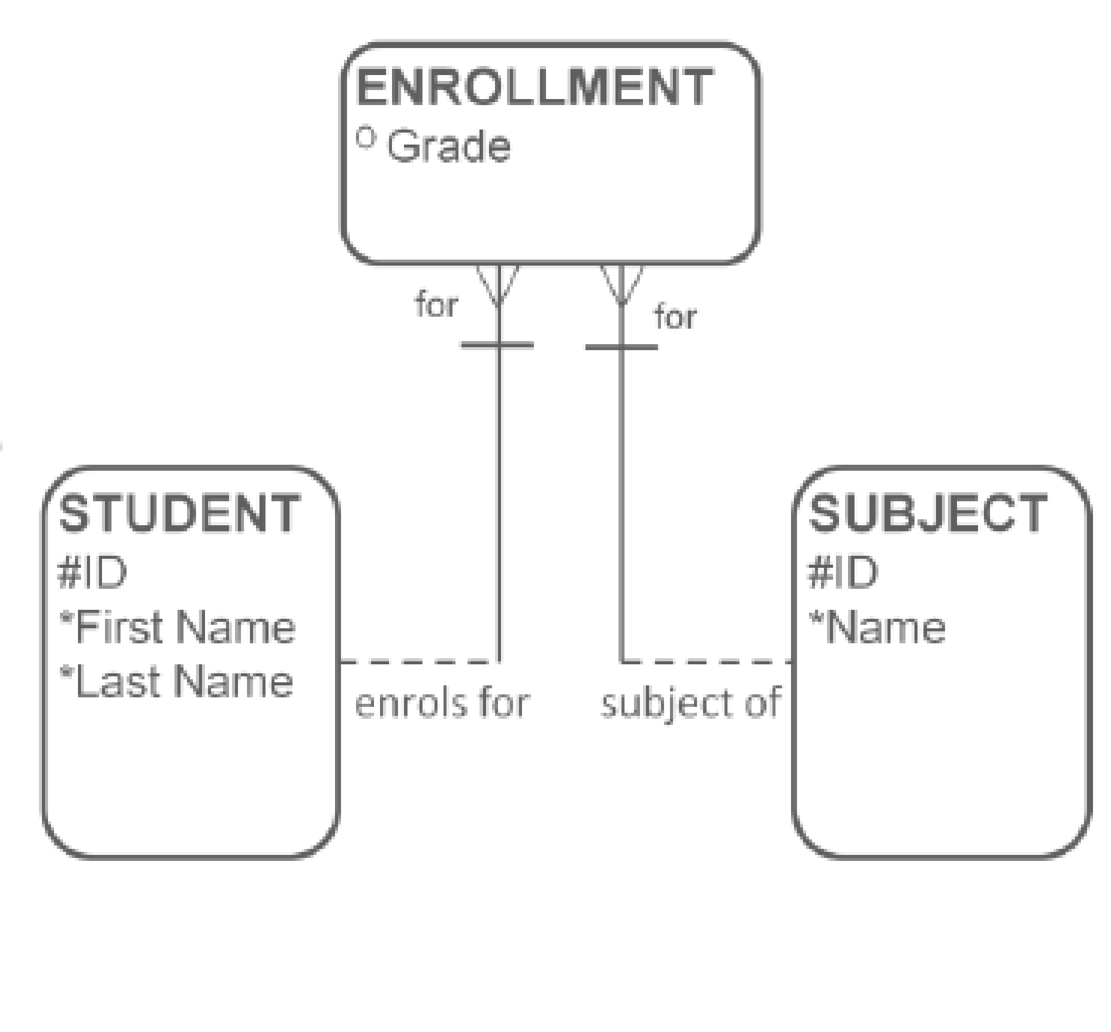
UIDs from Barred Relationships

- Sometimes the UID is a combination of an attribute and a relationship
- What is the UID of ACCOUNT? Is it artificial? Is it composite?
- Two people could have the same bank account number, but at different banks.
- Bank to bank transfers always need the bank routing number in addition to the account number.
- Now you need the IBAN number which is a composite key.


ACCOUNT
# number
* balance
* date opened
at
have
BANK
# number
* name
* location

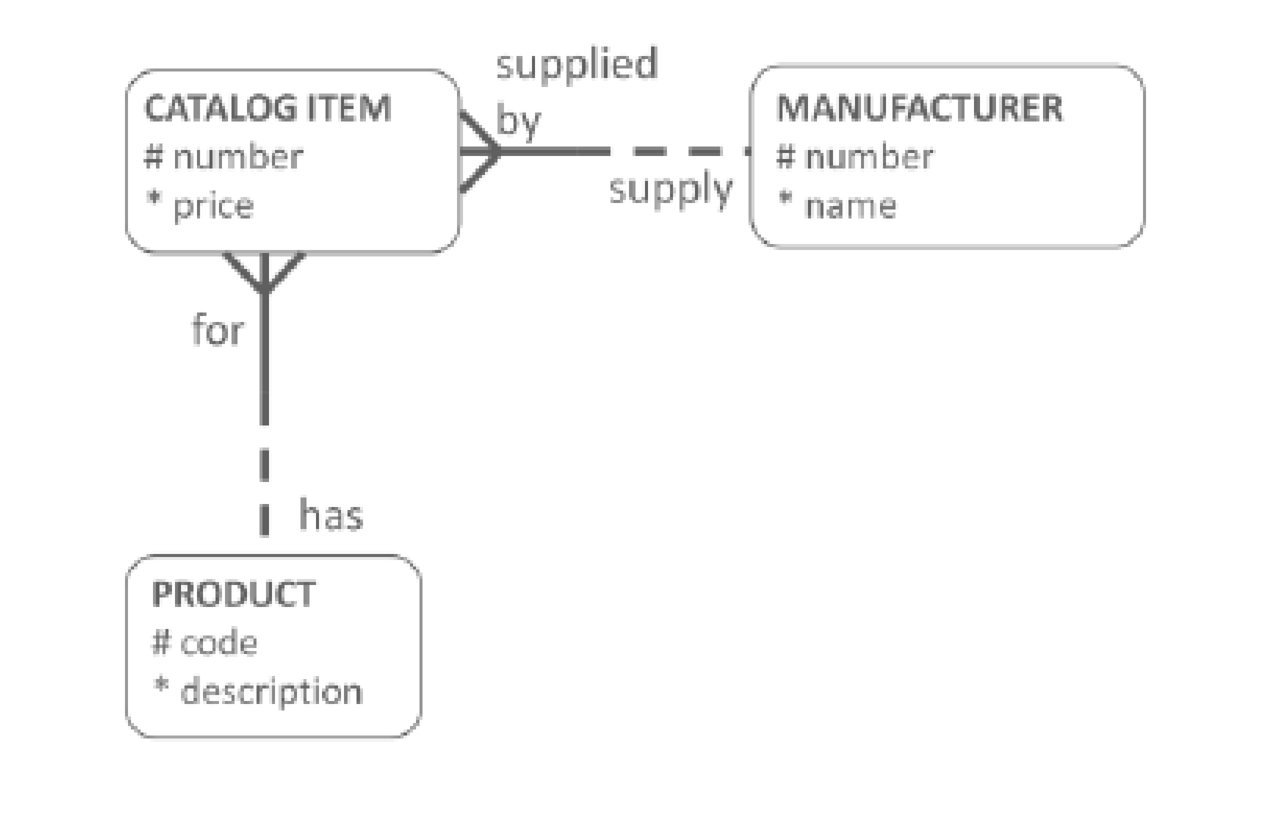
UID from Barred Relationship Intersection Entity

- As we’ve seen before the resolution of a M:M relationship often results in barred relationships from the intersection entity to the original ones.
- In this example, the UID of ENROLLMENT comes from STUDENT and SUBJECT



Artificial UID Intersection Entity

- It is possible for an intersection entity to use an artificial attribute as the UID, instead of the barred relationships to the original entities.



Each MANUFACTURER may produce one or more PRODUCTS

Each PRODUCT may be produced by one or more MANUFACTURERs

CATALOG ITEM resolves the M:M relationship

Artificial UID Intersection Entity

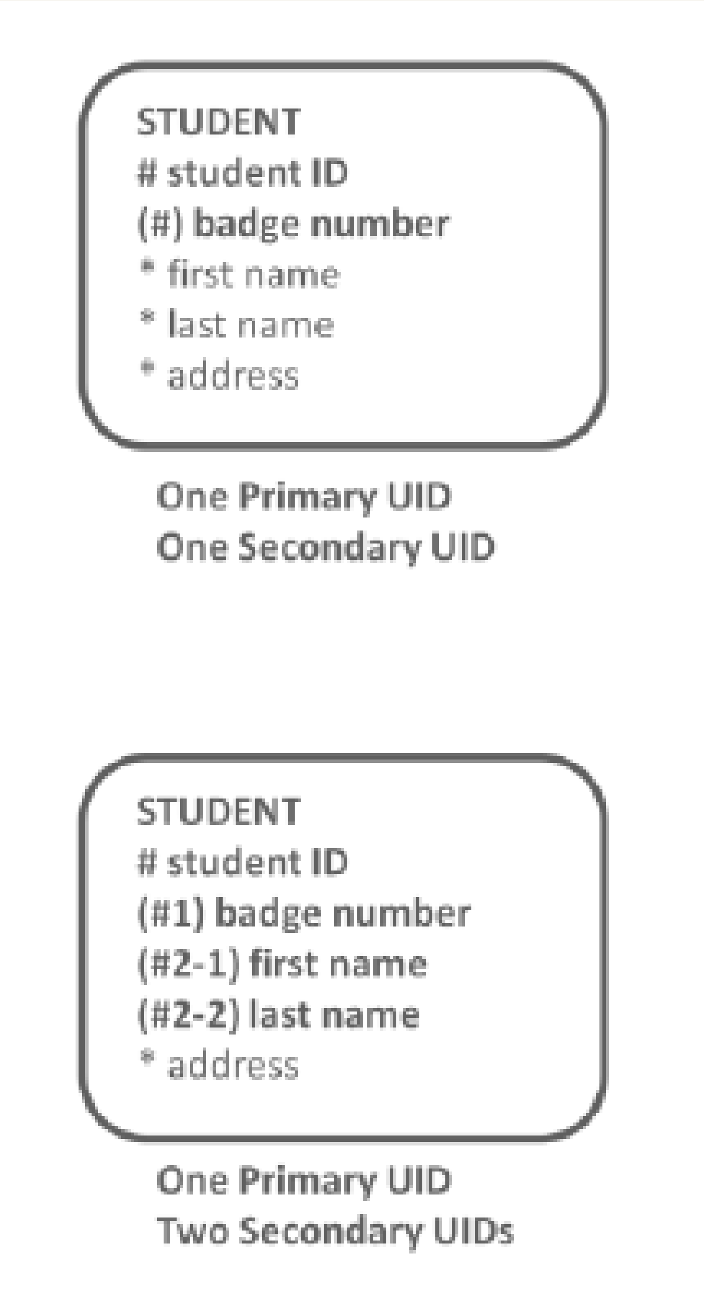
- CATALOG ITEM resolves the many to many relationship
- An item in a catalog could be uniquely identified by the manufacturer number and the product code, but the relationship is not barred, an artificial UID (catalog number) has been created instead.

Candidate UIDs

- Sometimes two or more possible UIDs exist
- For example, when you order a product from a commerical website, you will be assigned a unique customer code and asked to enter your email address
- Each uniquely identifies you, and each could be chosen as the UID. These are both considered candidate UIDs
- Only one candidate UID is chosen as the actual UID. This is called the primary UID.
- The other candidate UIDs are called secondary UIDs
- Secondary UIDs can be useful for searching in data.

Candidate UIDs

- Student ID has been chosen as the primary UID in both the STUDENT entities
- The first entity has one secondary UID, while the second has two secondary UIDs (one of which is composite)



STUDENT
# student ID
(#) badge number
* first name
* last name
* address

One Primary UID
One Secondary UID

STUDENT
# student ID
(#1) badge number
(#2-1) first name
(#2-2) last name
* address

One Primary UID
Two Secondary UIDs

Identification: Database vs. Real World

- Unique identifiers make it possible for us to distinguish one instance of an entity from another
- Eventually these become the primary keys in the database
- Primary keys in the database allow you to find specific records