BINF 8211/6211
Design and Implementation of Bioinformatics Databases
Lecture #6

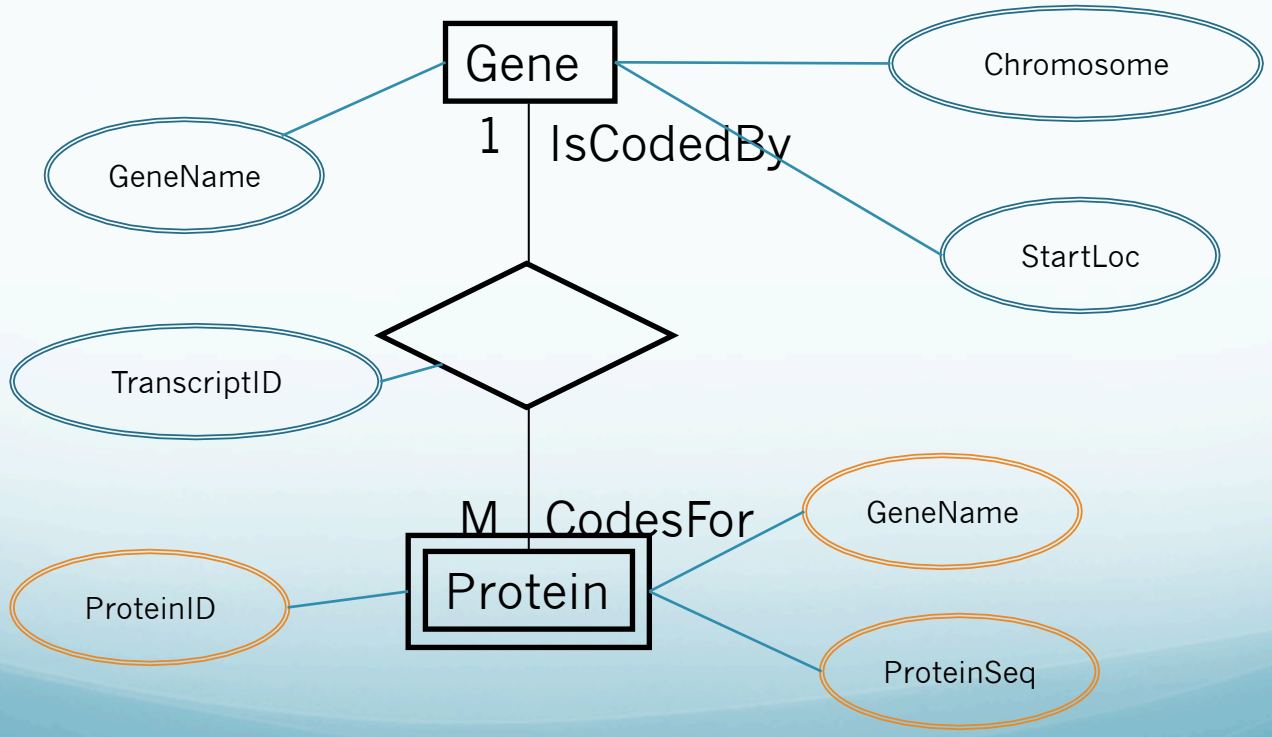
Dr. D. Andrew Carr
Dept. Bioinformatics and Genomics UNCC
Spring 2016

ER Modeling tools

- For this class
  - MySQL Workbench
  - http://dev.mysql.com/downloads/workbench/5.1.html
    - Open source
    - Free
    - Easy to use
- Other acceptable tools
  - Microsoft Visio
  - Oracle

Modeling Attributes

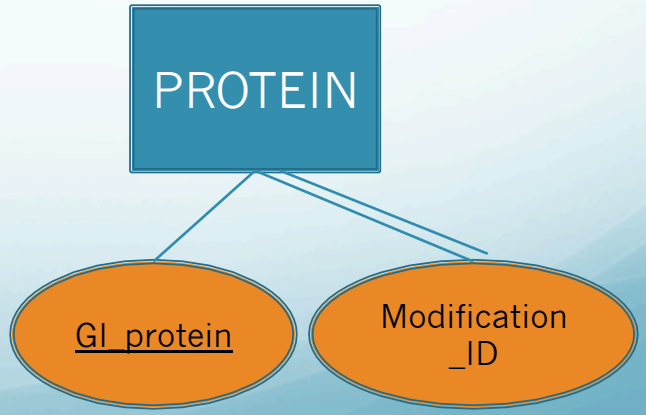
- In the IE example attributes were listed below the Entity Name. In the Chen style these are ovals that attach to the entity (gets messy).



Identifiers = KeyAttributes

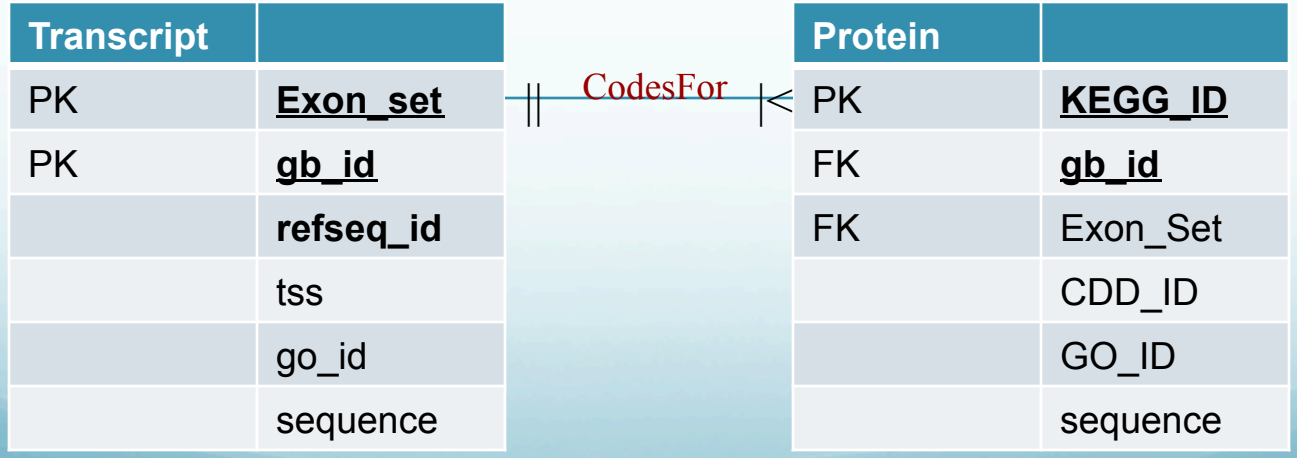
- Every entity instance ('row') must have something that uniquely identifies it – a key for retrieval.
  - The identifying attribute is
    - Presented in **bold** type (Not Null), is underlined or starred (designated as key) and is at the top of the attribute list.
    - In the data dictionary it is labeled as the primary key (PK), and some design programs place it in the graphical output as well
  - Composite key: several attributes provide a combination of values that together are unique (they must be the same attribute for all instances).

Transcript	
PK	**exon_set**
PK	**gb_id**
	refseq_id
	tss
	go_id

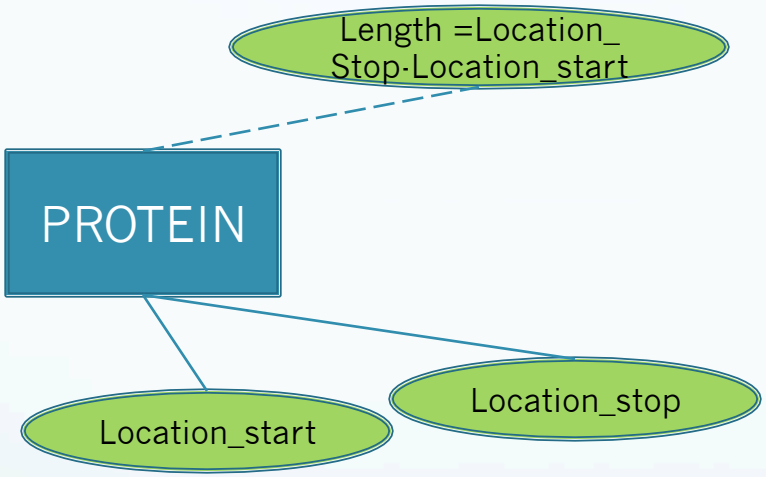


Foreign Key Attributes

- Controlled redundancy, through shared attributes, is how you line up rows in different entities that you want to combine and then filter.
  - One of the tables will be the parent, where the values are first entered and are non-redundant (the '1' side of a 1:M relationship). Note: just because it is non-redundant does not mean that it is the Primary Key, although it may be a candidate key.
  - The other table will be the child (the 'M' side) – in this case the attribute is labeled as a Foreign Key (FK) and there may be 0,1 to M rows.



Another type of attribute you can use is derived or computed, which can be done *as data is loaded* or *at the time of a query*.



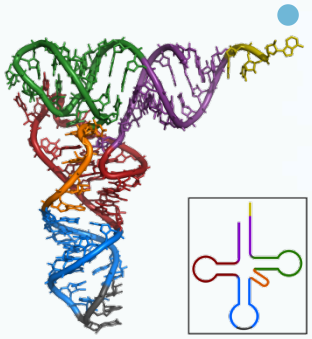
Advantages to storing value?

- CPU use minimized
- Record always there
- Query complexity is less

Disadvantages to storing value?

- Currency of data

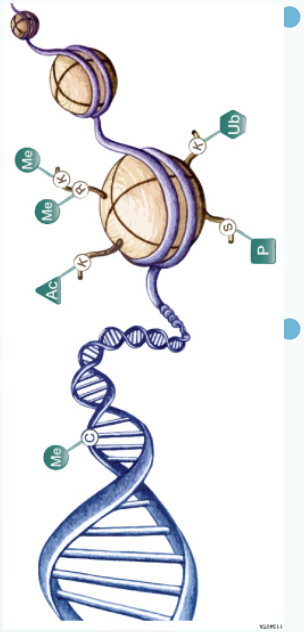
Attributes: Composition



- A simple attribute is atomic – you cannot break down the elements and retain any meaning.
  - Biomolecule Type: {Protein, Nucleic Acid, Carbohydrate, Lipid} CHAR type, domain is the list
  - Location (at time T): Cellular anatomy term, CHAR type, domain is in ontology list
  - A tRNA carries exactly one anti-codon complement

- A composite attribute has discrete chunks that have meaning
  - The GenBank Accession ID has the reference set, the accession number and may have a version number
    - NM_014313
  - A polypeptide may carry a leader sequence and the mature protein sequence
  - You might leave the composite values intact if you never want to filter data by the chunks
    - For publication authors you probably won't separate the address into parts
    - But - for patient records, an epidemiologist might want to group by city or state.

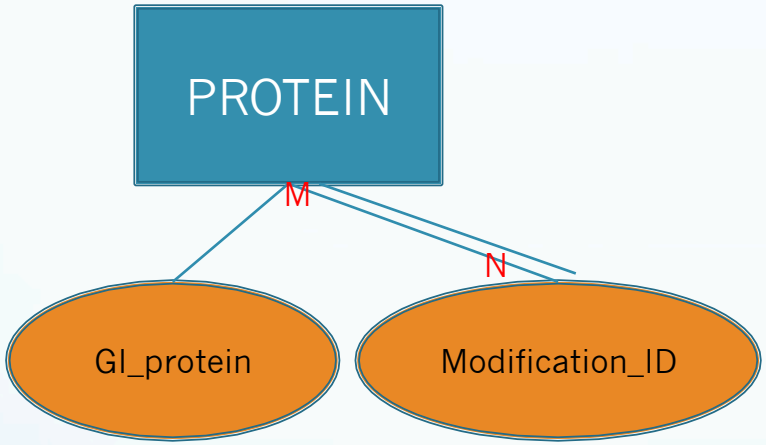
Attribute Values



- A *single-valued* attribute has been defined so that only one valid value can be assigned - it can be composite however.
  - The molecular weight of a polypeptide
  - The E.C. number of an enzyme ex EC 3.6.1.46: heterotrimeric G-protein GTPase
- A *multivalued attribute* can logically have more than one value – this usually means you have a M:N relationship that you have not resolved.
  - Protein modification - Histone 3 has a Lysine at position 4 that can have 1,2, or 3 methyl groups, all functionally relevant forms that can occur in a cell at the same time, hence the allowed values can be H3K0, H3K1, H3K2, H3K4.
  - To resolve the issue:
    - Decompose the attribute to 4 new attributes – then each can have a value (Present/Absent or concentration, for example)
      - You can end up with a lot of values of A, or '0' or NULL
      - If a new modification is discovered you will have to add yet more attributes
  - Create a new entity that specifically handles this set of attributes
    - ProteinMod (**PM_id**, **Protein_id**, ModificationMolecule, ModificationPosition)

Resolving the multi-valued attribute

Cell components example



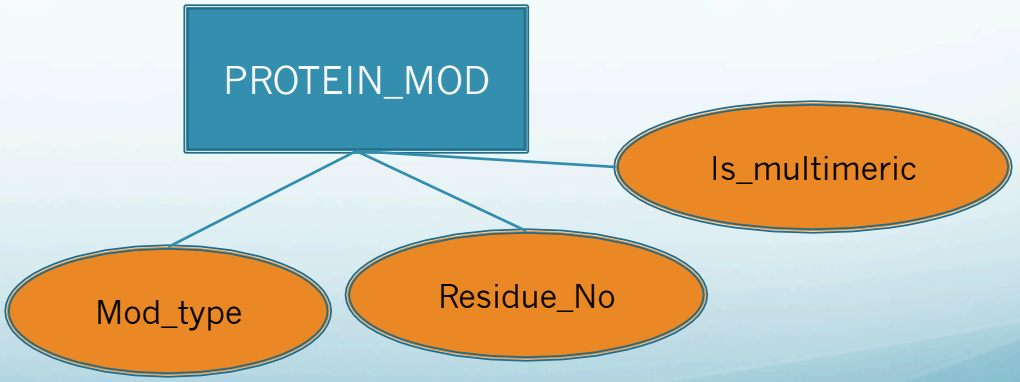
Protein
GI_Protein
Modification_ID

You could list all of the types of known modifications as attributes (multimeric, amino acid, etc)

Proteins without modifications will have a lot of NULLs. Newly discovered modifications will have to be added.

OR:

Create a new *entity* and relation, called Protein_modification



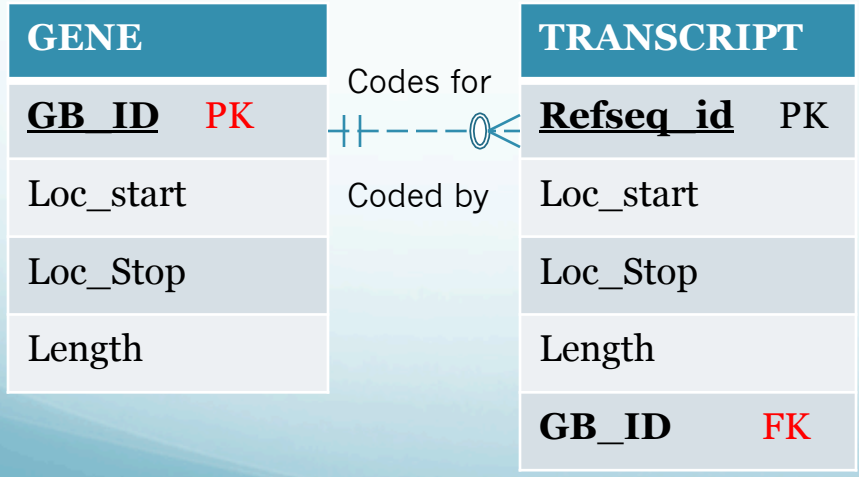
- What does a Bold attribute name mean in an IE model?
- What does an Underlined attribute name mean?
- Why is a primary key both bold and underlined?
- What is a simple attribute?
- What is a single-valued attribute?
- What is a composite attribute?
- What is a multi-valued attribute?
- Which of these must you adjust in a relational model and why?
- What is a derived attribute? Advantages and disadvantages?
- How many types of connectivity are there? Cardinality?

Quick Quiz questions

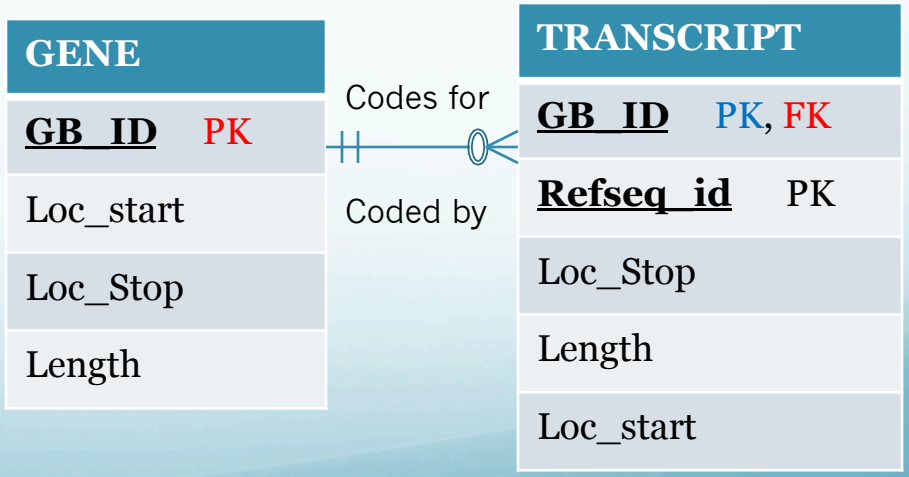
Relationship strength

- Non-identifying relationship (weak), does not use the PK of the parent entity as part of its own PK, but only as a FK
  - Diagram convention: Crowsfoot uses a dashed line (see below left)
- Identifying relationship (strong), when the PK of the related entity contains a PK component of the parent entity
- Implementation consequences: order of creation and data loading is essential.
  - You must load the 1 side first in a 1:M relationship to avoid referential integrity errors (weak or strong does not matter)

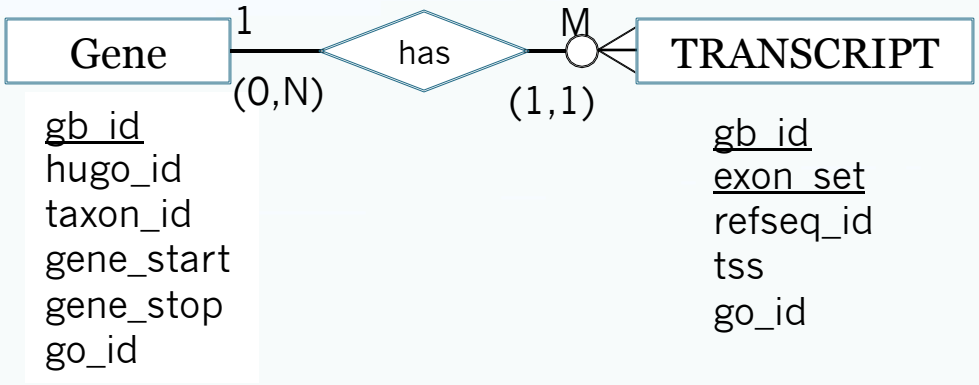
Weak (non-identifying) relationship



Strong (identifying) relationship

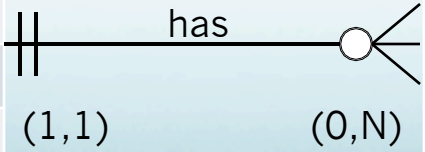


A *weak entity* is existence-dependent and has a primary key that is partially or totally derived from the parent entity in a relationship (strong relationship).



You can have an existence-dependent entity that is not weak because the PK is not defined using the PK of the related entity

Gene	
PK	**gb-id**
	hugo_id
	Taxon_id
	gene_start
	gene_stop
	go_id



Transcript	
PK	**exon_set**
PK, FK1	**gb_id**
	refseq_id
	tss
	go_id

The FK is mandatory because it is part of the PK; it's definition must include the specification that it is a FK that is NOT NULL.

Relationship Participation

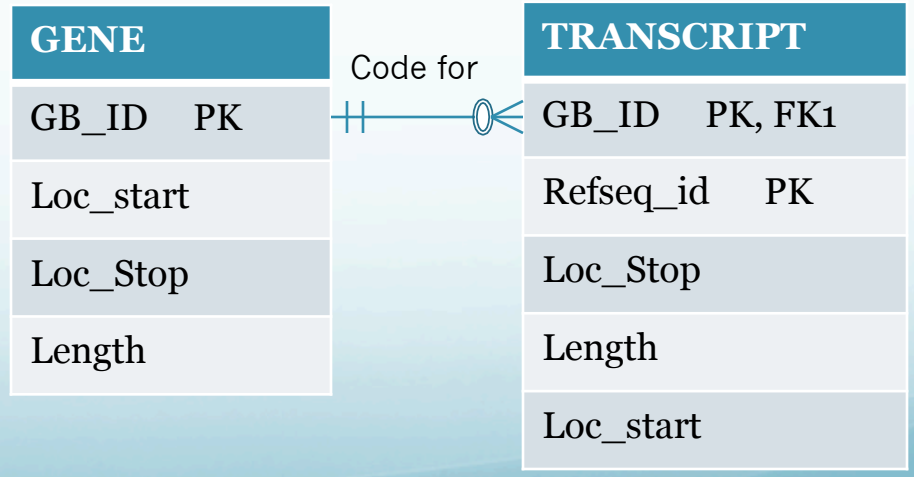
- When you define a relationship between two entity sets, the model can support the *possibility* of linked instances, or you may *require* the presence of linked instances.
  - Optional participation means what it says
    - Example – a polypeptide may have no post-translationally modified forms, so the existence of an instance in the ProteinModified entity set for an instance in the Polypeptide Set is Optional.
    - Graphically, use the 0< notation instead of the | < notation, since the minimum cardinality is zero instead of 1.
  - Mandatory participation also means what it says – there must be corresponding instances of entities across the relationship between two entity sets.
    - The minimum cardinality is 1, and the graphical notation is | < for the mandatory entity (or ||) .
    - The FK is NOT NULL

Relationship participation vs relationship strength

- Relationship participation and strength are **not** the same thing
  - The relationship is not necessarily weak when the entities are in an optional relationship nor strong when it is mandatory. For example:
    - Gene and transcript are in a strong relationship, but the transcript relationship is optional

- Relationship strength depends on how the PK is formulated, while participation depends on how the *process* is defined (rules).

Strong (identifying) relationship



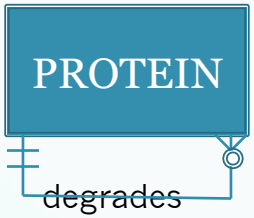
Relationship Degree

Categories in which an entity set has multiple relationships with other entity sets.

- Unary: there may be relationships between instances in the same entity set (this is recursive)
  - Example: a protease is an enzyme (a type of protein) whose substrate is other proteins – to query for the protein that is acted on by a protein requires referring to the protein table again.
- Binary: Two entity sets are associated – the most common design element
  - Example: one Gene has many Transcripts
- Ternary+: an entity set is on the 'Many' side of two or more other entity sets.
  - Example: A sample (Target) may be hybridized to many Microarrays, and a Microarray has many Probes. Microarray is on the 'M' side of both relations.

Unary Degree, *Recursive*

- Unary: one member of an entity has a relationship to itself or other members of the same set
  - Recursive; in ontology terms, the PART contains the PART. A unary relationship is always recursive.



If the relationship is 1:1, I can add a Substrate attribute whose values are the protein names. If it is 1:M (or M:M) I need a new entity.

Table Name: Protein

Protein_ID	Protein_Name	Substrate_Name
AAC63054	Thrombin	Fibrinogen
CAA33562	Fibrinogen	NONE

Table Name: Proteolytic Enzyme

Protein_ID	Enzyme_Name
DAA20347	Chymotrypsin
AGI80160	Elastase

proteolyzes

Table Name: ProteinTarget

Protein_ID	Protein_Name	Enzyme_Name
AAB59562	Keratin	Chymotrypsin
AAA98797	Albumin	Chymotrypsin
NP_000549	Hemoglobin, alpha subunit	Chymotrypsin
DAA20347	Chymotrypsin	Chymotrypsin
NT2NL_HUMAN	Notch Homolog2	Elastase

Higher Relationship Degrees

- Binary – the most common relationships are between pairs of entities.
- Ternary – to explain what we mean we require three entities
  - In a conceptual ERD these are usually decomposed into distinct binary representations.

TARGET

Sample_ID	Sample_Label	Array_ID	Expt_ID	RepNo	Label_ID	Sample_Species	Sample_Tissue
A1001	HUM_HeLa12	AffU133A	OC_Stage4	1	Cy3	Homo Sapiens	Tissue Culture
A1002	HUM_HeLa20	AffU133A	OC_Stage4	1	Cy5	Homo Sapiens	Tissue Culture
A1003	HUM_HeLa12	AffU133V2	OC_Stage4	2	Cy3	Homo Sapiens	Tissue Culture
A1004	HUM_HeLa20	AffU133V2	OC_Stage4	2	Cy5	Homo Sapiens	Tissue Culture

ARRAY

Array_ID	Expt_ID	Sample_ID	DataFilePath
AffU133A	OC_Stage4	A1001	Path
AffM10k	NakamuraLab314	A1001	path
AffU133V2	WellerLab610	A1002	path

PROBESET

ProbeSet_ID	Array_ID	Probe1Seq	Probe2Seq	Probe3Seq	Gene_ID
Hs_1541_s_22	AffU133A	AATTGTCCTG	TGGTAACCA	GAATGCCAA	NN2NL
Hs_1290_a_10	AffU133A	TATACCCAGG	TATCACTGT	CACGATTGA	APC1
Hs_0141_s_585	AffU133A	ACGACGCCACCA	GCAGACGTATAG	ATTATTCTCT	HARP2
Hs_2215_s_011	AffU133A	CGGCTAATTA	GAGATACA	GATTACCAA	SERC

Bridge Entities

- When you have a M:N relationship in your conceptual model, implementation will require that you decompose this to two 1:M relationships by adding another entity.
  - The new entity is called a Bridge or Composite Entity.
    - In spreadsheet land AKA the Linking Table.
  - The Bridge Entity PK is a composite of the PK of each parent entity set – when an attribute from another table it is called a Foreign Key. It this is the primary key of that table it will be unique and not null.
    - PK (FK1 NOT NULL, FK2 NOT NULL) where '1' is a named entity and '2' is a named entity.
    - There is an identifying relationship between the bridge entity and each parent (notation uses solid line).
    - The bridge entity is *existence-dependent* on the two parent entity sets.
    - The bridge entity can have additional attributes.

Linking Table example

Table Name: Enzyme

Protein_ID	Protein_Name
DAA20347	Chymotrypsin
AGI80160	Elastase

Table Name: ProteinTarget

Protein_ID	Protein_Name	Enzyme_Name
AAB59562	Keratin	Chymotrypsin
AAA98797	Albumin	Chymotrypsin
NP_000549	Hemoglobin, alpha subunit	Chymotrypsin
DAA20347	Chymotrypsin	Chymotrypsin
NT2NL_HUMAN	Notch Homolog2	Elastase

proteolyzes

Table Name: Enzyme

Protein_ID	Protein_Name
DAA20347	Chymotrypsin
AGI80160	Elastase

Table Name: Enzyme-ProteinTarget

PK, FK on Enzyme	**Protein_ID**
PK, FK on ProteinTarget	**Target_ID**
	CDD_ID

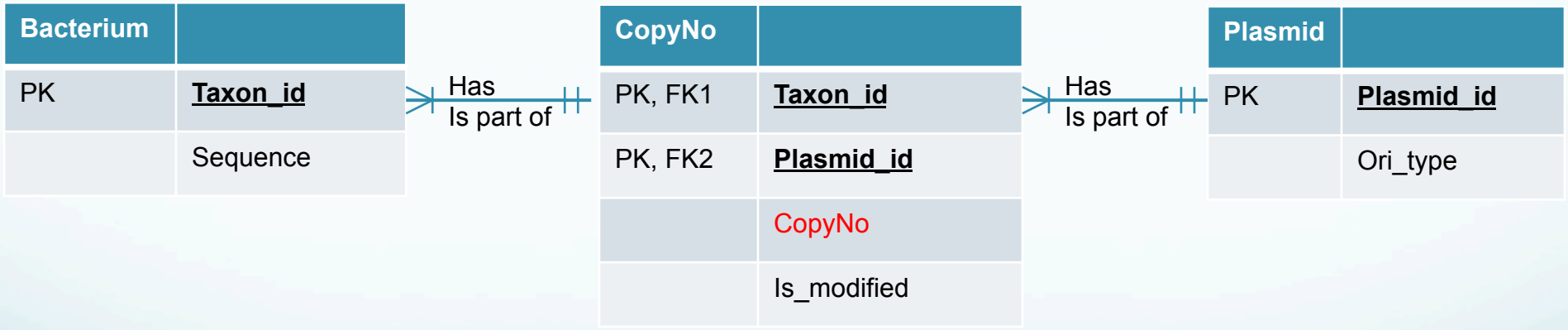
Table Name: ProteinTarget

Target_ID	Protein_Name	Enzyme_Name
AAB59562	Keratin	Chymotrypsin
AAA98797	Albumin	Chymotrypsin
NP_000549	Hemoglobin, alpha subunit	Chymotrypsin
DAA20347	Chymotrypsin	Chymotrypsin
NT2NL_HUMAN	Notch Homolog2	Elastase

Composite Entities - more examples

- Problem 1: In the relational model only entities have attributes, the relationships do not.
  - If you have microorganisms and plasmids the number of plasmids present *depends on* which microorganism is the host: the copy number is not absolute to either the plasmid or the microorganism but to the relationship between them
- Problem 2: the M:N relationship between two entities (each instance in each table has to line up to multiple instances in the other table – you need an indexing list)
  - Example might be relationship between Locus:Allele. There are many genetic loci in an organism, and there are many alleles that can occur at each locus. Each individual can have only as many alleles as it has chromosomes in somatic cells (bacteria have one, humans have 2, wheat has 6, potatoes have 8).

The composite entity creates a bridge or link between two entities that have a M:N relationship.

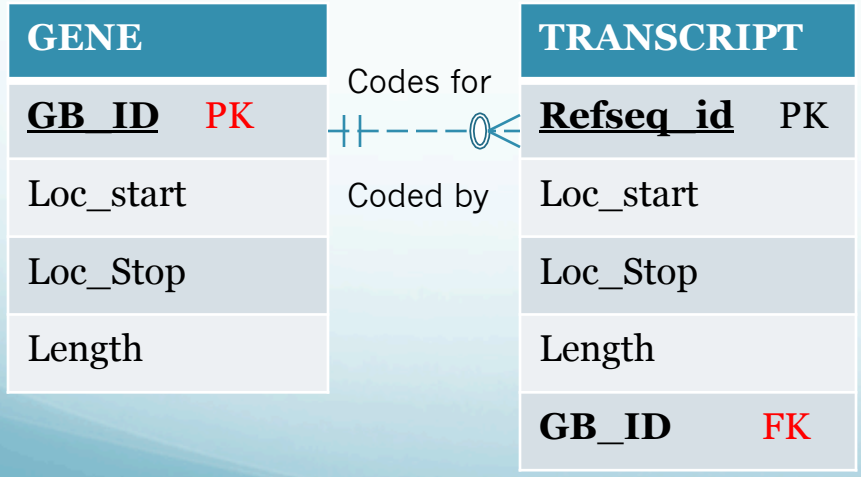


- Design step: a third entity is added that has a 1:M relationship with each of the others.

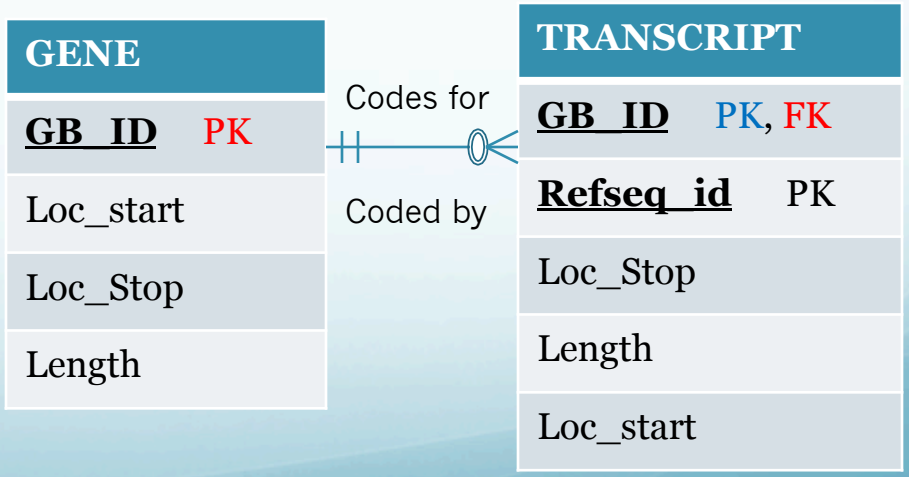
Relationship *strength*

- Non-identifying relationship (weak), does not use the PK of the parent entity as part of its own PK, but only as a FK
  - Diagram convention: Crow's foot uses a dashed line (see below left)
- Identifying relationship (strong), when the PK of the related entity contains a PK component of the parent entity
- Implementation consequences: order of creation and data loading is essential.
  - You must load the 1 side first in a 1:M relationship to avoid referential integrity errors (weak or strong does not matter)

Weak (non-identifying) relationship

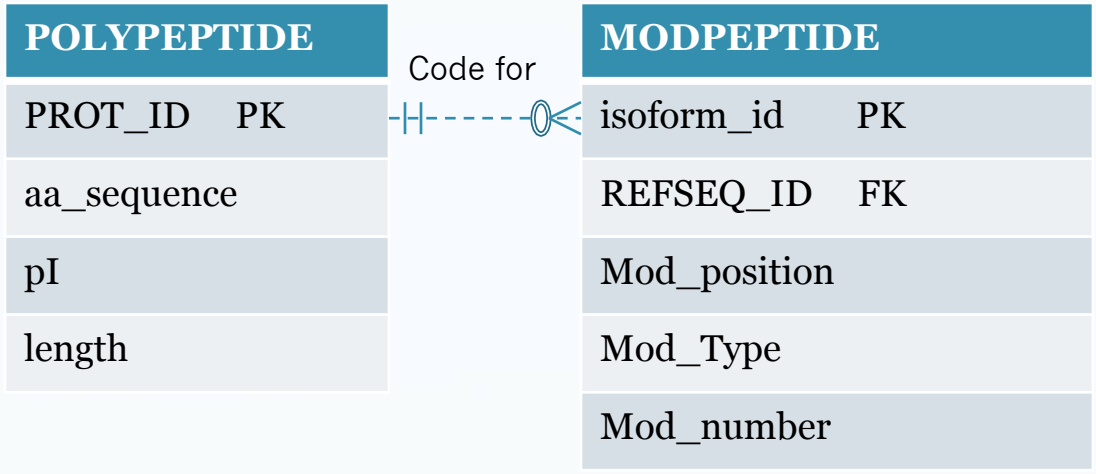


Strong (identifying) relationship

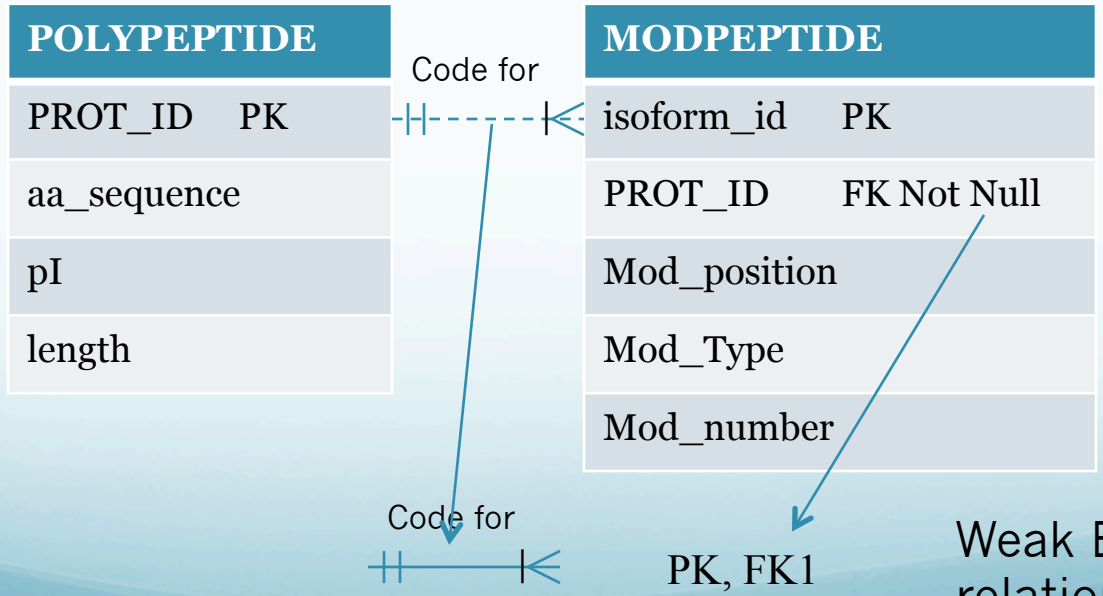


Relationship Participation

- When you define a relationship between two entity sets, the model can support the *possibility* of linked instances, or you may *require* the presence of linked instances.
  - Optional participation means what it says – there can be an instance in either entity without a match the other.
    - Example – a polypeptide may have no post-translationally modified forms, so the existence of an instance (next slide) in the ProteinModified entity set for an instance in the Polypeptide set is Optional.
      - The FK in a child table is allowed to be NULL
      - The optional side shows a zero for range
    - Graphically, use the 0< notation instead of the |< notation, since the minimum cardinality is zero instead of 1.
  - Mandatory participation also means what it says – there must be corresponding instances of entities across the relationship between two entity sets.
    - The minimum cardinality is 1, and the graphical notation is | < for the mandatory entity (or ||) .
    - The FK in the child table is NOT NULL
    - The mandatory side has a ‘1 in the diagram



Optional participation



Mandatory participation

Weak Entity, strong relationship

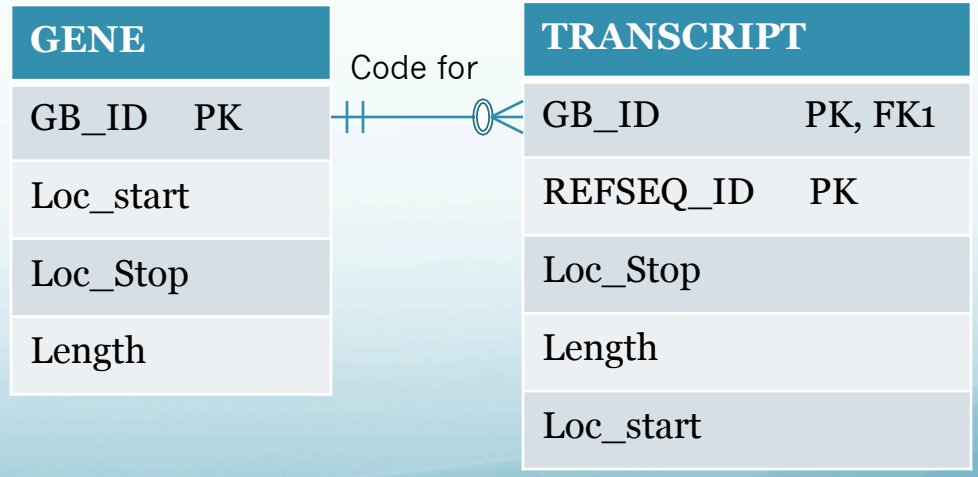
Relationship participation vs relationship strength

Relationship participation and strength are **not** the same thing: the relationship is not necessarily weak when the entities are in an optional relationship nor strong when it is mandatory.

Relationship strength (strong/weak) depends on how the PK is formulated, while *participation* (mandatory/optional) depends on how the *process* is defined (rules).

Entity strength depends on both participation and strength: a weak entity has mandatory participation and a strong relationship (PK of parent defines entity).

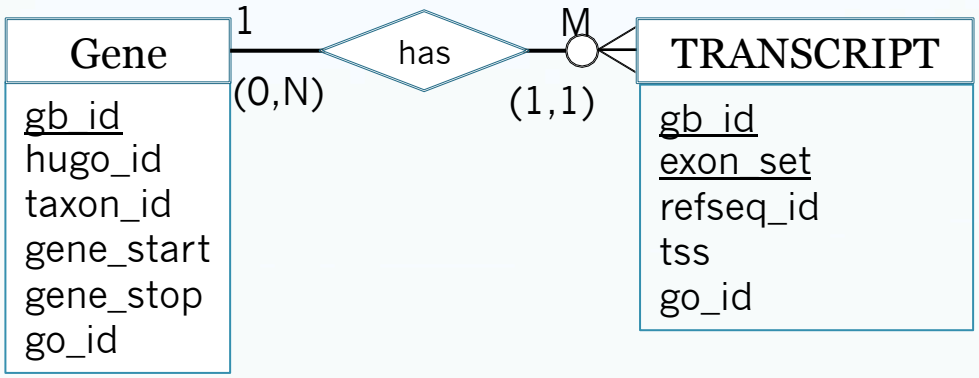
Strong (identifying) relationship



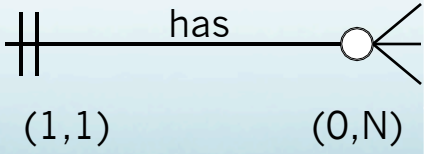
Graphical Notation Summary and also a check-list for figuring out the categories.

Relationship	FK - child side	Domain - child side	IE connector
Optional	NULL	(0…N)	-0<
Mandatory	NOT NULL	(1…N)	-|<
Weak	Single PK		- - - - - - -
Strong	Composite PK		----------
Entity			
Weak	Composite PK	(1…N)	- - - |<

Test Yourself



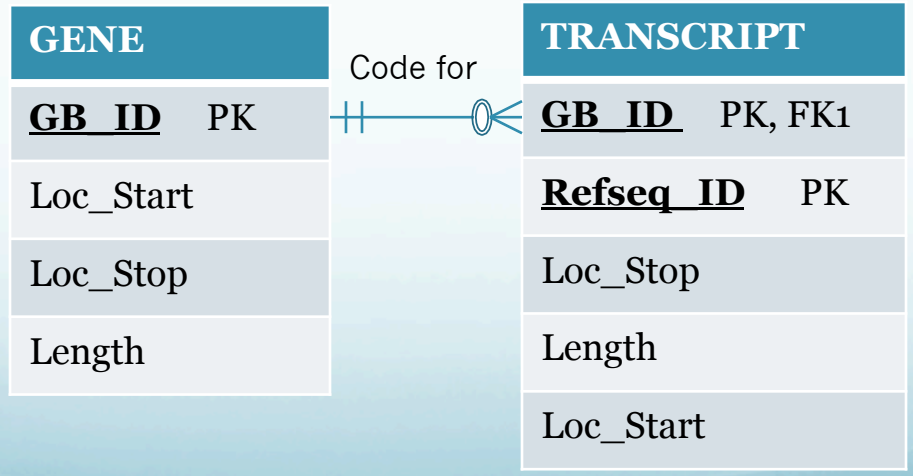
Gene	
PK	**gb-id**
	hugo_id
	Taxon_id
	gene_start
	gene_stop
	go_id



Transcript	
PK	**exon_set**
PK, FK1	**gb_id**
	refseq_id
	tss
	go_id

A shorthand notation for text communication

- Graphical notation is a good shorthand way to communicate a lot of information
  - It is slow to encode and requires special software
- Text notation convention for entities:
  - GENE(**GB_ID,**Loc_Start, Loc_Stop, Length**),** TRANSCRIPT(**GB_ID**,**Refseq_ID**, Loc_Stop, Length, Loc_Start**)**
- Why can I have the same attribute name in both entities?



Indexing: in large tables, data retrieval can be quite slow. Indexes are a kind of Table of Contents to a database, they can be generated automatically by the DBMS, or the administrator can direct their creation.

GENE	
Refseq_id	symbol
NG_0089723.1	HCN2
NG_0000523.2	SLC34A2
NG_0899664.1	FOX4A

How can I find all of the alternative transcripts known for the gene HCN2?

Method 1: Scan TRANSCRIPT for NG_089723.1

TRANSCRIPT		
Transcript_id	Transcript_symbol	Refseq_id
AC_4366875	HCN2.3	NG_0089723.1 *
AC_8824311	FOX4A.2	NG_0899664.1
AC_3455769	HCN2.5	NG_0089723.1 *
AC_7778112	HCN2.1	NG_0089723.1 *

Method 2: pre-scan the table and organize the records

TRANSCRIPT		
Transcript_id	Transcript_symbol	Refseq_id
AC_4366875 *	HCN2.3	NG_0089723.1 *
AC_8824311	FOX4A.2	NG_0899664.1
AC_3455769 *	HCN2.5	NG_0089723.1 *
AC_7778112 *	HCN2.1	NG_0089723.1 *

NG_0089723.1 maps to rows 1, 3 and 4 so I can use AC_4366875, AC_3455769 and AC_7778112 as *the index keys* and store them with pointers to rows 1, 3 and 4.

It is allowed to index on one *or more* attributes in a given table.

Most DBMS ***automatically index the PKs*** of each table, as a unique index key with a single –row pointer.

File types and meanings

- FastQ files: Output from NGS sequencing instruments
  - Capillary sequencers produced .abi files or .fas files to show how and electropherogram was interpreted as a base string
- Standard Flowgram Files (SFF) files
- SAM/BAM files
  - BAM is the binary form of a SAM file (Sequence Alignment/Map format)
    - A tab-delimited text file with sequence alignment data
    - Indexes the read by genomic position
    - http://samtools.sourceforge.net/

FastQ files

- A text-based format for storing the base call and the quality score for a DNA sequence together. Extension is usually .fq or .fastq

- Generally you will see 4 lines/read
  - @ with an identifier and a description that is optional
  - The base calls as letters
  - A '+' and possibly the identifier repeated (optional)
  - The quality values, encoded as single ASCI characters (note that @ and + are *used in the quality string* so you have to be sure you are on the correct line).

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=36
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=36
IIIIIIIIIIIIIIIIIIIIIIIIIIIII9IG9IC
```

Examples in these slides are taken from Wikipedia, Sourceforge, SEQanswers and http://wiki.christophchamp.com/index.php/FASTQ_format

Sanger FastQ

- The first line is descriptive and arbitrarily long – SRR is a Sequence Read archive identifier – this will ALWAYS be present in NCBI files.
  - The 071112_SLXA-EAS1 is another identifier indicating the name of the platform – Solexa was bought by Illumina.
  - EAS1 etc provides meta-data about the instrument (next slides)
  - Length tells you how many base calls and quality scores to expect.

- Add 33 to the actual Phred Score, then select the ASCII character in this position. A range of 0-93 is allowed.
  - Characters 33-126 are allowed.
  - Illumina has used a number of encoding schemes and special characters in different versions, so be sure to pay attention to the version and check the manual before combining data.

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=36
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=36
IIIIIIIIIIIIIIIIIIIIIIIIIIIII9IG9IC
```

```
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
 |                         |    |        |                              |                     |
33                        59   64       73                            104                   126
 0........................26...31.......40
```

Illumina FastQ

- The ouput from Illumina platforms provides a lot of information in the identifier.

```
@HWUSI-EAS100R:6:73:941:1973#0/1
```

HWUSI-EAS100R	the unique instrument name
6	flowcell lane
73	tile number within the flowcell lane
941	'x'-coordinate of the cluster within the tile
1973	'y'-coordinate of the cluster within the tile
#0	index number for a multiplexed sample (0 for no indexing)
/1	the member of a pair, /1 or /2 *(paired-end or mate-pair reads only)*

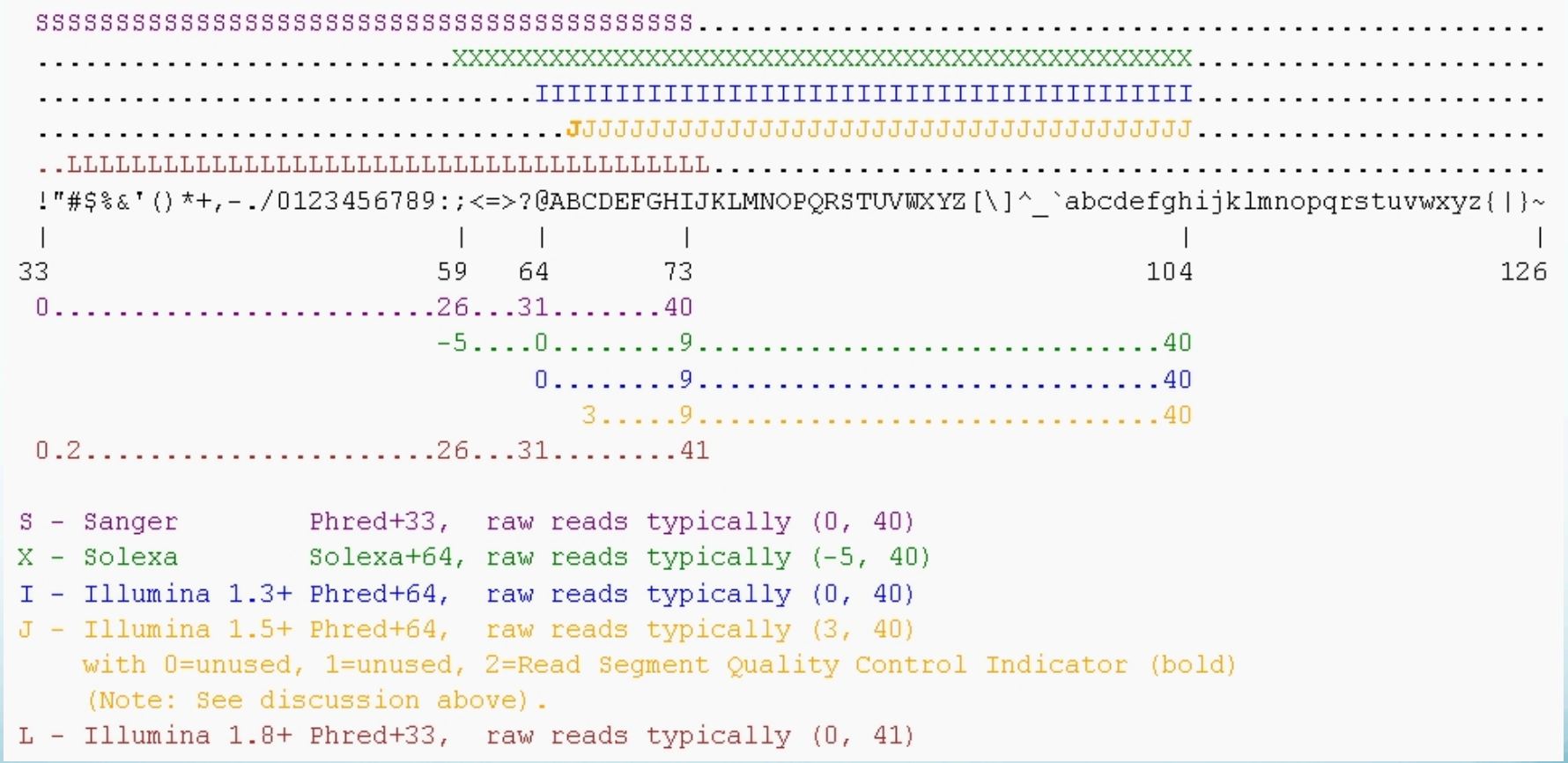
Versions of the Illumina pipeline since 1.4 appear to use **#NNNNNN** instead of **#0** for t

With Casava 1.8 the format of the '@' line has changed:

```
@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG
```

EAS139	the unique instrument name
136	the run id
FC706VJ	the flowcell id
2	flowcell lane
2104	tile number within the flowcell lane
15343	'x'-coordinate of the cluster within the tile
197393	'y'-coordinate of the cluster within the tile
1	the member of a pair, 1 or 2 *(paired-end or mate-pair reads only)*
Y	Y if the read is filtered, N otherwise
18	0 when none of the control bits are on, otherwise it is an even number
ATCACG	index sequence

Q-value encodings



Sanger format encodes a Phred quality score from 0-93 using ASCII 33-126, as does SAM

Homework #2: Entities

- We have 6 entity sets: gene, transcript, proteins, modified proteins, modified protein activity and modified protein localization (make sure there are at least 5 entity sets).

- The attributes used will include at least 3 each (they must at least cover the key attributes)
  - GENE (**Gene_ID**, **Gene_name**, MaximumCDS)
  - TRANSCRIPT (**Transcript_id**, **Gene _Name, Transcript_Sequence**)
  - PROTEIN (**Protein_ID**, Transcript_ID, Protein_sequence)
  - MOD_PROTEIN (**ModProtein_ID**, Protein_ID, ModProtein_Type)
  - MODPROTEIN_ACTIVITY (**ModProtein_ID** ,EC_No, Activity_Name,)
  - MODPROTEIN_LOCATION (**ModProtein_ID, Location_ID**, CellCycleStage)

Homework #2: Relationship Definitions

- Business rules will include the following
  - Each gene can produce zero to many transcripts
  - Each transcript comes from one and only one gene and produces exactly one protein
  - A protein sequence can be either measured or inferred.
  - A transcript can exist without a gene to which it maps
  - Each protein is produced by one and only one transcript.
  - A protein can exist without a transcript to which it maps.
  - A protein can have zero to many modified amino acids
  - A protein can be modified on more than one amino acid
  - A protein is a type of modified protein (type none) - recursive
  - Amino acids may be modified by more than one type of group.
  - A modified protein is existence-dependent on a protein
  - Proteins can be located at multiple locations in the cell

Homework #2

- Using MySQL Workbench
  - Create and ER IE model of the given entities
  - The model should include the relationships covering the rules defined above.
  - Due Thursday Feb 4 at 8:00 a.m.
  - Turn in the saved model file.