Relational Databases for Biologists

Session 2
SQL To Data Mine
A Database

Robert Latek, Ph.D.
Sr. Bioinformatics Scientist
Whitehead Institute for Biomedical Research

Session 2 Outline

- Database Basics
- Review E-R Diagrams And db4bio
- Data Types And Values
- Connecting To MySQL
- Relational Algebra
- Data Mining SQL

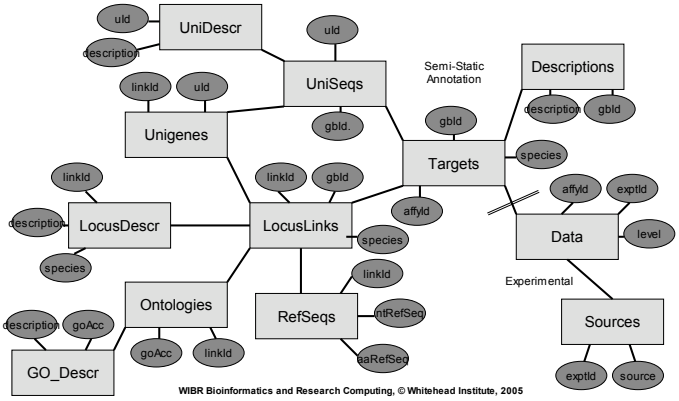
Database Basics

- Databases Are Composed Of Tables (Relations)
- Relations Are Entities That Have Attributes (Column Labels) And Tuples (Records)
- Databases Can Be Designed From E-R Diagrams That Are Easily Converted To Tables
- Primary Keys Uniquely Identify Individual Tuples And Represent Links Between Tables

Building An E-R Diagram

- Identify Data Attributes
- Conceptualize Entities By Grouping Related Attributes
- Identify Relationships/Links
- Draw Preliminary E-R Diagram
- Add Cardinalities And References
- Refine E-R Diagram By Applying Design Principles

db4bio E-R Diagram II



Number Data Types

- **INT**
  - Signed -2147483648 to 2147483647
  - Unsigned 1844674407370551615
- **FLOAT/DOUBLE[(M,D)]**
  - Decimal values, 1.234, 1.47564839E+5
  - M is display size, D is number of decimals
- **DATE/DATETIME**
  - '1000-01-01 00:00:00' to '9999-12-31 23:59:59'
  - 'YYYY-MM-DD HH:MM:SS'
- **TIMESTAMP**
  - YYYYMMDDHHMMSS

Character Data Types

- **VARCHAR(M)**
  - M characters is length, Text up to 255 characters
  - VARCHAR(5)
    - Will store Apple as 'Apple'
    - Will store Pineapple as 'Pinea'
- **TEXT**
  - Text up to 65535 characters

- VARCHARs and TEXTs must always be described inside of quotes, single or double
  - Food = "Apple"

Data Values

- **NULL vs. NOT NULL**
  - Data can either require a value for each tuple or not need one.
- **KEY**
  - Primary keys must be NOT NULL
- **Default**
  - If an attribute was specified as NULL its default is automatically NULL (characters) or empty (numbers).
  - If an attribute was specified as NOT NULL its default value is automatically "" (characters) or zero (numbers).
  - The default value can also be specified manually.

Connecting To MySQL

- If No Local MySQL, In Terminal Window
  - % ssh hebrides.wi.mit.edu -l username
- Connect to MySQL Database Server
  - % mysql -u username -p -D db4bio
  - mysql>

- SQL Commands Are Case-Insensitive
- Tables And Attributes Are Case-Sensitive

> DESCRIBE Table;

- > DESCRIBE Data;

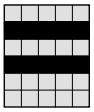
```
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| affyId | varchar(30) |      | PRI |         |       |
| exptId | varchar(10) |      | PRI |         |       |
| level  | int(11)     |      |     | 0       |       |
+--------+-------------+------+-----+---------+-------+
```

- > DESCRIBE LocusDescr;

```
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| linkId      | int(11)      |      | PRI | 0       |       |
| description | varchar(100) | YES  |     | NULL    |       |
| species     | varchar(20)  | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
```

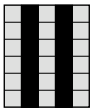

Relational Algebra

- Restrict: Remove Tuples That Don't Fit a Specific Criteria.



Restrict

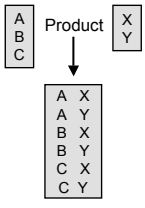
- Project: Remove Specific Attributes



Project

Table Product And Divide

- Product: Merge Tuples From Two Tables In Every Possible Way



- Divide: Separate Tuples That Have Every Tuple In Another Table

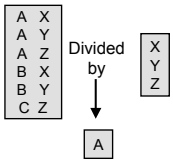


Table Join

- Similar To Product Except That Merged Tuples Must Satisfy A Specific Requirement

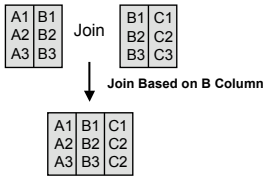
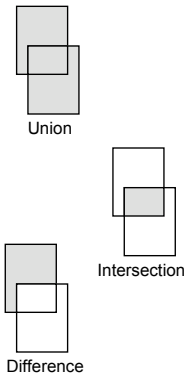


Table Algebra

- Union: Combine Tuples From Both Tables Without Duplicates
- Intersection: Remove Tuples That Are Not Found In Both Tables
- Difference: Remove Tuples That Are Not Shared In One Of The Tables



Aggregates

- Aggregates Act On An Attribute
  - AVG()
    - AVG(level)
  - COUNT()
    - COUNT(affyId)
  - MAX()
    - MAX(level)
  - MIN()
    - MIN(species)
  - SUM()
    - SUM(level)

Project

- List Nucleotide RefSeqs In RefSeqs Table

```
+--------+-----------+-----------+
| linkId | ntRefSeq  | aaRefSeq  |
+--------+-----------+-----------+
|      1 | NM_130786 | NP_570602 |
|      2 | NM_000014 | NP_000005 |
|      3 | NG_001067 |           |
|      9 | NM_000662 | NP_000653 |
|     10 | NM_000015 | NP_000006 |
+--------+-----------+-----------+
```

```
> SELECT ntRefSeq
  FROM RefSeqs
  LIMIT 5;
```

```
+-----------+
| ntRefSeq  |
+-----------+
| NM_130786 |
| NM_000014 |
| NG_001067 |
| NM_000662 |
| NM_000015 |
+-----------+
```


Project With Math

- List Expression Levels And Twice Level In Data Table

```
+-----------------+----------+-------+
| affyId          | exptId   | level |
+-----------------+----------+-------+
| AFFX-MurIL2_at  | hs-cer-1 |    20 |
| AFFX-MurIL10_at | hs-cer-1 |     8 |
| AFFX-MurIL4_at  | hs-cer-1 |    77 |
| AFFX-MurFAS_at  | hs-cer-1 |    30 |
| AFFX-BioB-5_at  | hs-cer-1 |   258 |
+-----------------+----------+-------+
```

```
> SELECT level, level*2
  FROM Data
  LIMIT 5;
```

```
+-------+---------+
| level | level*2 |
+-------+---------+
|    20 |      40 |
|     8 |      16 |
|    77 |     154 |
|    30 |      60 |
|   258 |     516 |
+-------+---------+
```


Restrict

- List All Human Tuples in Targets

```
+-----------+----------+---------+
| affyId    | gbId     | species |
+-----------+----------+---------+
| 100_g_at  | Y08200   | Hs      |
| 1000_at   | X60188   | Hs      |
| 100000_at | AI846313 | Mm      |
| 100001_at | M18228   | Mm      |
| 100002_at | X70393   | Mm      |
+-----------+----------+---------+
```

```
> SELECT *
  FROM Targets
  WHERE species="Hs"
  LIMIT 5;
```

```
+-----------+--------+---------+
| affyId    | gbId   | species |
+-----------+--------+---------+
| 100_g_at  | Y08200 | Hs      |
| 1000_at   | X60188 | Hs      |
| 1001_at   | X60957 | Hs      |
| 1002_f_at | X65962 | Hs      |
| 1003_s_at | X68149 | Hs      |
+-----------+--------+---------+
```

Using WHERE

- Restricts Queries Having Lists, Ranges, Inequalities, Patterns

```
> SELECT ntRefSeq, aaRefSeq
  FROM RefSeqs
  WHERE linkId = 10;
```

ntRefSeq	aaRefSeq
NM_000015	NP_000006

```
> SELECT *
  FROM RefSeqs
  WHERE linkId LIKE "105%"
  LIMIT 5;
```

linkId	ntRefSeq	aaRefSeq
1050	NM_004364	NP_004355
1051	NM_005194	NP_005185
1052	NM_005195	NP_005186
1053	NM_001805	NP_001796
1054	NM_001806	NP_001797

Using WHERE

```
> SELECT *
  FROM GO_Descr
  WHERE description = "collagen";
```

goAcc	description
GO:0005202	collagen
GO:0005581	collagen

```
> SELECT *
  FROM Data
  WHERE affyId = "1000_at";
```

affyId	exptId	level
1000_at	hs-cer-1	960
1000_at	hs-hrt-1	441
1000_at	hs-liv-1	744

Using ORDER BY

- Lists Results In Numerical/Alphabetical Order According To Specified Tuples

```
> SELECT *
  FROM RefSeqs
  WHERE linkId LIKE "105%"
  ORDER BY linkId DESC;
```

linkId	ntRefSeq	aaRefSeq
105910	NM_134094	NP_598855
105892	XM_128276	XP_128276
105887	XM_127943	XP_127943
105870	XM_128254	XP_128254
105866	XM_128271	XP_128271

```
> SELECT *
  FROM RefSeqs
  WHERE linkId LIKE "105%"
  ORDER BY aaRefSeq DESC;
```

linkId	ntRefSeq	aaRefSeq
105021	XM_147708	XP_147708
105493	XM_139194	XP_139194
105003	XM_138081	XP_138081
105756	XM_128277	XP_128277
105892	XM_128276	XP_128276

Advanced WHERE

```
> SELECT affyId, level
  FROM Data
  WHERE level between 80 and 100
  LIMIT 5;
```

affyId	level
AFFX-BioB-3_at	97
AFFX-HUMTFRR/M11507_3_at	90
AFFX-HSAC07/X00351_M_st	86
31324_at	91
31356_at	91

```
> SELECT *
  FROM UniSeqs
  WHERE gbId
  NOT LIKE "NM_%" LIMIT 5;
```

uId	gbId
Hs.2	D90042
Hs.4	X03350
Hs.11	D90278
Hs.11	L00693
Hs.21	M16652

Mining With WHERE

```
> SELECT *
  FROM Data
  WHERE level between 80 and 100
  OR level < 21
  LIMIT 5;
```

affyId	exptId	level
AFFX-MurIL2_at	hs-cer-1	20
AFFX-MurIL10_at	hs-cer-1	8
AFFX-BioB-3_at	hs-cer-1	97
AFFX-BioB-5_st	hs-cer-1	20
AFFX-BioB-M_st	hs-cer-1	20

```
> SELECT affyId, level
  FROM Data
  WHERE exptId != "hs-cer-1"
  AND level BETWEEN 250 AND 300
  LIMIT 5;
```

affyId	level
AFFX-M27830_3_at	271
AFFX-HUMGAPDH/M33197_3_st	277
31315_at	250
31362_at	256
31510_s_at	257

Using GROUP BY

- Operates Only On The Tuples That Were Not Removed By a Where

```
> SELECT *
  FROM Data
  WHERE level < 100
  GROUP BY level,affyId LIMIT 5;
```

affyId	exptId	level
100054_s_at	mm-liv-1	1
100325_at	mm-hrt-1	1
100435_at	mm-cer-1	1
100547_at	mm-hrt-1	1
100988_at	mm-cer-1	1

```
> SELECT *
  FROM Data
  WHERE level < 100
  GROUP BY affyId, level LIMIT 5;
```

affyId	exptId	level
100001_at	mm-hrt-1	5
100001_at	mm-cer-1	20
100002_at	mm-hrt-1	20
100003_at	mm-hrt-1	20
100006_at	mm-liv-1	68

Using HAVING

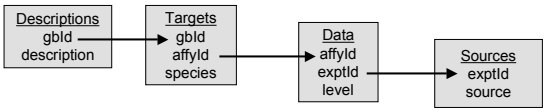
- Sets The Conditions For the GROUP BY Clause Like WHERE Sets Conditions For SELECT
- CAN Use Aggregates

```
> SELECT description
  FROM GO_Descr
  GROUP BY description
  HAVING COUNT(description)>1
  LIMIT 5;
```

```
+-----------------------------------------------------+
| description                                         |
+-----------------------------------------------------+
| 1-phosphatidylinositol-4-phosphate kinase, class I  |
| 2-oxo-delta3-4,5,5-trimethylcyclopentenylacetyl-Co  |
| 3-methyl-2-oxobutanoate dehydrogenase (lipoamide)   |
| actin modification                                  |
| activation of transcription on exit from mitosis,   |
+-----------------------------------------------------+
```

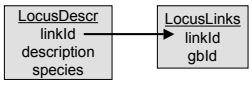

Natural Joins

- Table Joining Links Tables Together Through Their Relationships And Allows You To Traverse Your Schema/Database
- Use SELECT And FROM To Join Tables
- Join Through Common Attributes With WHERE And AND Using Theta Operators: =,<,>,!=,>=, <=
- Traverse From Descriptions To Sources



Binary Table Join

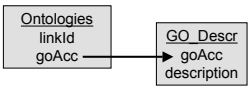
```
> SELECT LocusDescr.description, LocusDescr.species, LocusLinks.gbId
  FROM LocusDescr, LocusLinks
  WHERE LocusDescr.linkId = LocusLinks.linkId
  GROUP BY LocusLinks.gbId
  LIMIT 5;
```




```
+-------------------------------------------+---------+--------+
| description                               | species | gbId   |
+-------------------------------------------+---------+--------+
| granulysin                                | Hs      | A00142 |
| lipase, gastric                           | Hs      | A01046 |
| serine (or cysteine) proteinase inhibitor | Hs      | A03911 |
| albumin                                   | Hs      | A06977 |
| S100 calcium binding protein A8           | Hs      | A12027 |
+-------------------------------------------+---------+--------+
```


Binary Table Join

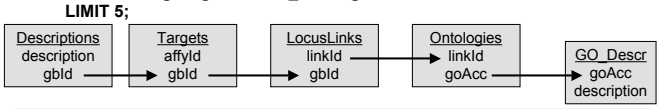
```
> SELECT GO_Descr.description, Ontologies.linkId
  FROM GO_Descr, Ontologies
  WHERE Ontologies.goAcc=GO_Descr.goAcc
  LIMIT 5;
```




```
+---------------------------------+--------+
| description                     | linkId |
+---------------------------------+--------+
| protein carrier                 |      2 |
| arylamine N-acetyltransferase   |      9 |
| arylamine N-acetyltransferase   |     10 |
| serine protease inhibitor       |     12 |
| enzyme                          |     13 |
+---------------------------------+--------+
```


Multiple Table Join

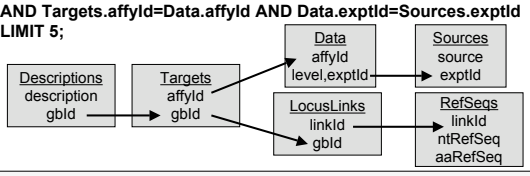
```
> SELECT Descriptions.description AS gene_description,
  GO_Descr.description AS GO_description
  FROM Descriptions, GO_Descr, LocusLinks, Ontologies, Targets
  WHERE Descriptions.gbId=Targets.gbId
  AND Targets.gbId=LocusLinks.gbId
  AND LocusLinks.linkId=Ontologies.linkId
  AND Ontologies.goAcc=GO_Descr.goAcc
  LIMIT 5;
```




```
+-----------------------------------------+-----------------------------------------+
| gene_description                        | GO_description                          |
+-----------------------------------------+-----------------------------------------+
| HSLFBPS7 Human fructose-1,6-biphosphatase | fructose-2,6-bisphosphate 2-phosphatase |
| HSU30872 Human mitosin mRNA, complete cds | regulation of mitosis                   |
| HSU33052 Human lipid-activated, protein kinase | protein kinase                    |
| HSU33053 Human lipid-activated protein kinase | protein kinase                    |
| HSU33920 Human clone lambda 5 semaphorin mRNA, | extracellular space                |
+-----------------------------------------+-----------------------------------------+
```


Mega Table Join

```
> SELECT Descriptions.description, Sources.source, RefSeqs.ntRefSeq
  FROM Descriptions, Sources, RefSeqs, Targets, LocusLinks, Data
  WHERE Descriptions.gbId=Targets.gbId
  AND Targets.gbId=LocusLinks.gbId
  AND LocusLinks.linkId=RefSeqs.linkId
  AND Targets.affyId=Data.affyId AND Data.exptId=Sources.exptId
  LIMIT 5;
```

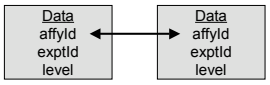



```
+-----------------------------------------------------------+-------------+-----------+
| description                                               | source      | ntRefSeq  |
+-----------------------------------------------------------+-------------+-----------+
| Homo sapiens immunoglobulin lambda gene locus DNA, clone:288A10   | Human liver | NG_000002 |
| HSIGVL009 Human rearranged immunoglobulin lambda light chain mRNA | Human heart | NG_000002 |
| Homo sapiens immunoglobulin lambda gene locus DNA, clone:31F3     | Human liver | NG_000002 |
| Homo sapiens immunoglobulin lambda gene locus DNA, clone:288A10   | Human brain | NG_000002 |
| HSIGVL009 Human rearranged immunoglobulin lambda light chain mRNA | Human liver | NG_000002 |
+-----------------------------------------------------------+-------------+-----------+
```

Table Self Join

- Identify Relationships Between Data Within A Single Table

```
> SELECT Data1.affyId, Data1.exptId as exptId1, Data2.exptId as exptId2,
  Data1.level as level1, Data2.level as level2
  FROM Data Data1, Data Data2
  WHERE Data1.affyId=Data2.affyId
  AND Data1.level >= Data2.level*2
  LIMIT 5;
```



affyId	exptId1	exptId2	level1	level2
AFFX-MurIL10_at	hs-cer-1	hs-hrt-1	8	4
AFFX-MurIL4_at	hs-cer-1	hs-hrt-1	77	20
AFFX-BioB-M_at	hs-cer-1	mm-cer-1	214	20
AFFX-BioB-M_at	hs-cer-1	mm-hrt-1	214	48
AFFX-BioB-M_at	hs-cer-1	mm-liv-1	214	20

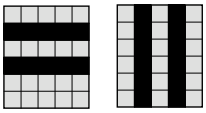
Master Table Self Join

```
> SELECT Data1.affyId, Data1.exptId as exptId1, Data2.exptId as exptId2,
  Data1.level as level1, Data2.level as level2
  FROM Data Data1, Data Data2
  WHERE Data1.affyId=Data2.affyId
  AND Data1.level BETWEEN Data2.level*2 AND Data2.level*3
  ORDER BY Data1.affyId
  LIMIT 5;
```

affyId	exptId1	exptId2	level1	level2
100014_at	mm-hrt-1	mm-liv-1	52	20
100014_at	mm-cer-1	mm-liv-1	55	20
100015_at	mm-cer-1	mm-hrt-1	943	396
100015_at	mm-cer-1	mm-liv-1	943	468
100024_at	mm-hrt-1	mm-liv-1	306	111

Summary

- Tables Store Data Of Specific Types
- Data Can Have Default Values And Be NOT NULL Restricted
- Restrict And Project



- Use WHERE Or HAVING To Constrain SELECT
- Table Joins Highlight The Relationships Between Data In A Database

Next Session

- Build Your Own Database!
- Use SQL To CREATE Tables And Specify Their Structure
- Use SQL To INSERT and DELETE Data Into Your Database
- Use SQL To UPDATE/Modify Your Database
- Input Data Files Directly Into Your Database

Exercise Your SQL