



Relational Database Queries

BMI 535/635

SQL Query

```
mysql> SELECT * FROM Gene LIMIT 10;
```

GeneId	Name	Biotype	Chr	Start	End	Strand
ENSG00000223972	DDX11L1	transcribed_unprocessed_pseudogene	1	11869	14409	+
ENSG00000227232	WASH7P	unprocessed_pseudogene	1	14404	29570	-
ENSG00000278267	MIR6859-1	miRNA	1	17369	17436	-
ENSG00000243485	MIR1302-2HG	lincRNA	1	29554	31109	+
ENSG00000284332	MIR1302-2	miRNA	1	30366	30503	+
ENSG00000237613	FAM138A	lincRNA	1	34554	36081	-
ENSG00000268020	OR4G4P	unprocessed_pseudogene	1	52473	53312	+
ENSG00000240361	OR4G11P	transcribed_unprocessed_pseudogene	1	57598	64116	+
ENSG00000186092	OR4F5	protein_coding	1	65419	71585	+
ENSG00000238009	AL627309.1	lincRNA	1	89295	133723	-

```
10 rows in set (0.00 sec)
```

SQL Query Syntax

```
SELECT [DISTINCT] select-list  
FROM from-list  
[WHERE qualification]
```

from-list: list of tables

select-list: column names named in the from-list

qualification: boolean combination of conditions

DISTINCT: returned rows should not contain
duplicates [OPTIONAL]

Basic SQL Query

Every query

- must have a SELECT clause
 - specifies columns of interest
- must have a FROM clause
 - specifies cross-product of tables
- optional WHERE clause
 - selection conditions on tables listed in FROM

```
SELECT [DISTINCT] select-list  
FROM from-list  
[WHERE qualification]
```

SQL Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

```
7 rows in set (0.00 sec)
```

List the names of genes?

SELECT [DISTINCT] select-list
FROM from-list
WHERE qualification

SQL Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> SELECT Name FROM Gene LIMIT 10;
```

Name
TSPAN6
TNMD
DPM1
SCYL3
C1orf112
FGR
CFH
FUCA2
GCLC
NFYA

10 rows in set (0.00 sec)

SQL Query: DISTINCT

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> SELECT DISTINCT Name FROM Gene LIMIT 10;
```

Name
TSPAN6
TNMD
DPM1
SCYL3
C1orf112
FGR
CFH
FUCA2
GCLC
NFYA

10 rows in set (0.00 sec)

SQL Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

List all information for FGFR2?

SQL Query: WHERE

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

List all information for FGFR2?

```
mysql> SELECT * FROM Gene WHERE Name='FGFR2';
```

GeneId	Name	Biotype	Chr	Start	End	Strand
ENSG00000066468	FGFR2	protein_coding	10	121478334	121598458	-

1 row in set (0.05 sec)

SQL Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

```
7 rows in set (0.00 sec)
```

List all information for FGFR2 and TP53?

SQL Query: OR

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

List all information for FGFR2 and TP53?

```
mysql> SELECT * FROM Gene WHERE Name='FGFR2' OR Name='TP53';
```

GeneId	Name	Biotype	Chr	Start	End	Strand
ENSG000000066468	FGFR2	protein_coding	10	121478334	121598458	-
ENSG00000141510	TP53	protein_coding	17	7661779	7687550	-

2 rows in set (0.04 sec)

SQL Query: UNION

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

List all information for FGFR2 and TP53?

```
mysql> SELECT * FROM Gene WHERE Name='FGFR2'  
-> UNION  
-> SELECT * FROM Gene WHERE Name='TP53';
```

GeneId	Name	Biotype	Chr	Start	End	Strand
ENSG00000066468	FGFR2	protein_coding	10	121478334	121598458	-
ENSG00000141510	TP53	protein_coding	17	7661779	7687550	-

2 rows in set (0.07 sec)

SQL Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

List all genes on the “-” strand and are protein_coding?

SQL Query: AND

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> SELECT Name FROM Gene WHERE Strand='-' AND Biotype='protein_coding' LIMIT 10;
```

Name
TSPAN6
DPM1
SCYL3
FGR
FUCA2
GCLC
STPG1
LAS1L
CYP51A1
KRIT1

10 rows in set (0.00 sec)

SQL Query: Nested Queries

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> SELECT Name FROM Gene WHERE Strand='-' IN  
-> (SELECT Strand FROM Gene WHERE Biotype='protein_coding') LIMIT 10;
```

Name
TSPAN6
DPM1
SCYL3
FGR
FUCA2
GCLC
STPG1
LAS1L
CYP51A1
KRIT1

10 rows in set (0.04 sec)

SQL Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

List all transcript ids of FGFR2?

SQL Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> DESCRIBE Transcript;
```

Field	Type	Null	Key	Default	Extra
TranscriptId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
SupportLevel	int(11)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.01 sec)

```
mysql> DESCRIBE TranscriptGene;
```

Field	Type	Null	Key	Default	Extra
TranscriptId	char(15)	NO	PRI	NULL	
GeneId	char(15)	NO	PRI	NULL	

2 rows in set (0.00 sec)

List all transcript ids of FGFR2?

SQL Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> DESCRIBE TranscriptGene;
```

Field	Type	Null	Key	Default	Extra
TranscriptId	char(15)	NO	PRI	NULL	
GeneId	char(15)	NO	PRI	NULL	

2 rows in set (0.00 sec)

List all transcript ids of FGFR2?

SQL Query: Nested Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> DESCRIBE TranscriptGene;
```

Field	Type	Null	Key	Default	Extra
TranscriptId	char(15)	NO	PRI	NULL	
GeneId	char(15)	NO	PRI	NULL	

2 rows in set (0.00 sec)

```
mysql> SELECT TranscriptId FROM TranscriptGene WHERE GeneId IN
-> (SELECT GeneId FROM Gene WHERE Name='FGFR2');
```

TranscriptId
ENST000000336553
ENST000000346997
ENST000000351936
ENST000000356226
ENST000000357555
ENST000000358487
ENST000000359354
ENST000000360144
ENST000000369056
ENST000000369058
ENST000000369059

SQL Query: INNER JOIN

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> DESCRIBE TranscriptGene;
```

Field	Type	Null	Key	Default	Extra
TranscriptId	char(15)	NO	PRI	NULL	
GeneId	char(15)	NO	PRI	NULL	

2 rows in set (0.00 sec)

```
mysql> SELECT t.TranscriptId FROM TranscriptGene t
-> INNER JOIN Gene g ON t.GeneId=g.GeneId
-> WHERE g.Name='FGFR2';
```

TranscriptId
ENST00000336553
ENST00000346997
ENST00000351936
ENST00000356226
ENST00000357555
ENST00000358487
ENST00000359354
ENST00000360144
ENST00000369056
ENST00000369058
ENST00000369059

SQL Query: INNER JOIN

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> DESCRIBE TranscriptGene;
```

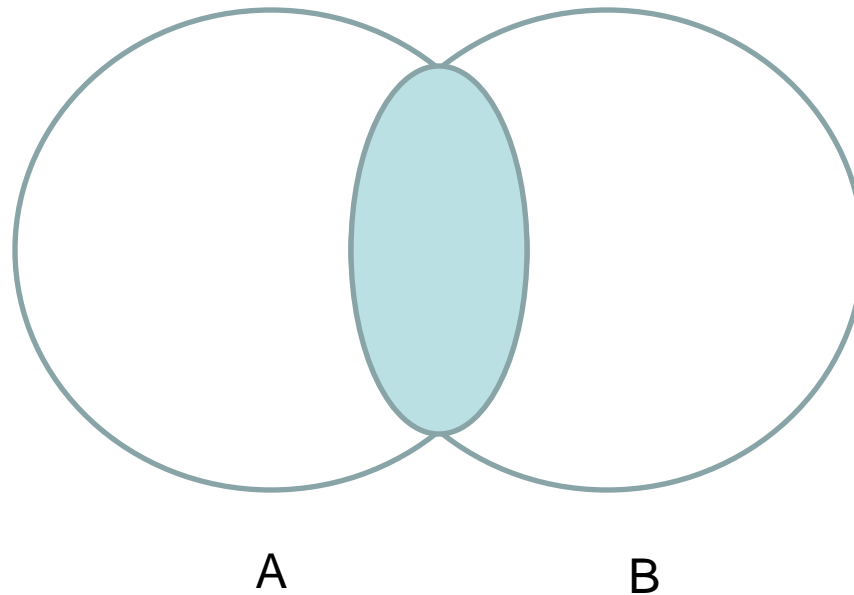
Field	Type	Null	Key	Default	Extra
TranscriptId	char(15)	NO	PRI	NULL	
GeneId	char(15)	NO	PRI	NULL	

2 rows in set (0.00 sec)

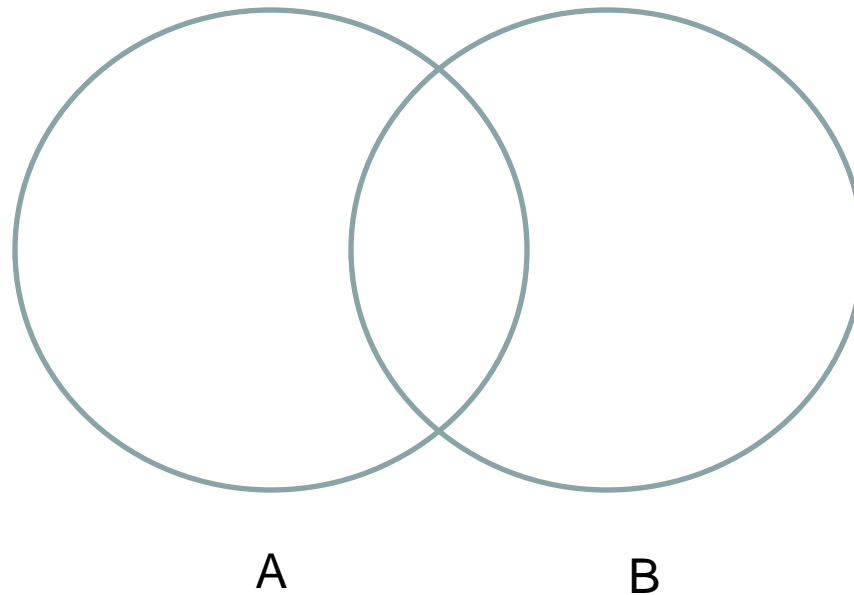
```
mysql> SELECT t.TranscriptId FROM TranscriptGene t, Gene g  
-> WHERE t.GeneId=g.GeneId AND g.Name='FGFR2';
```

TranscriptId
ENST00000336553
ENST00000346997
ENST00000351936
ENST00000356226
ENST00000357555
ENST00000358487
ENST00000359354
ENST00000360144
ENST00000369056
ENST00000369058
ENST00000369059

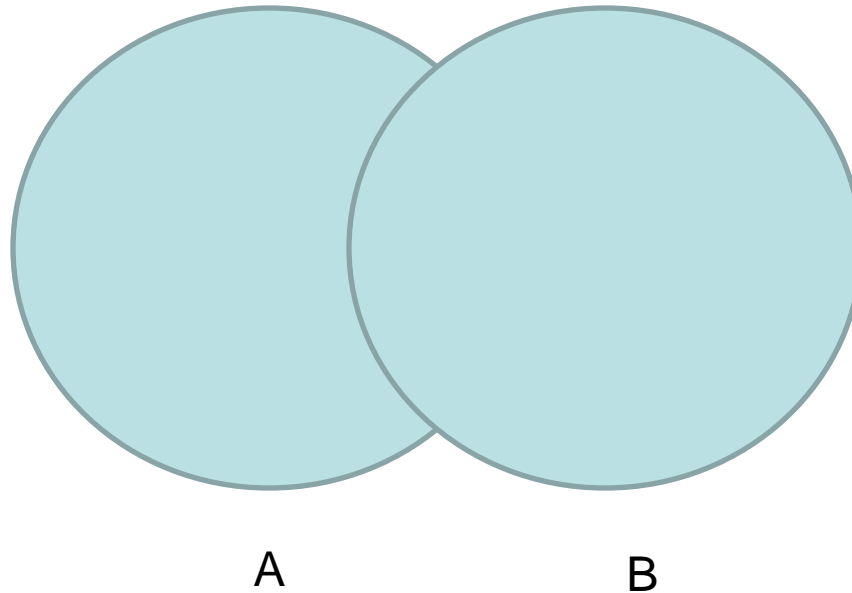
SQL Query: INNER JOIN



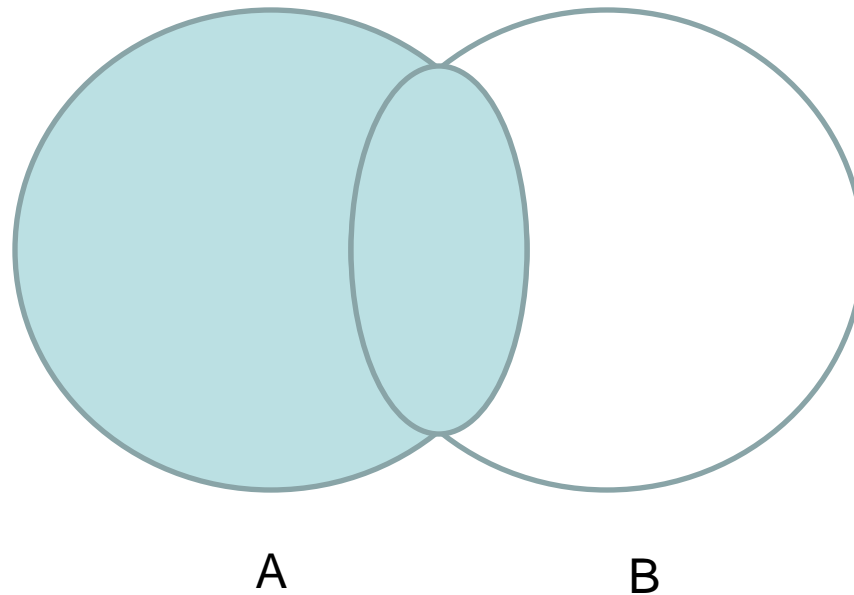
SQL Query: OUTER JOIN?



SQL Query: OUTER JOIN



SQL Query: LEFT OUTER JOIN



SQL Query: RIGHT OUTER JOIN

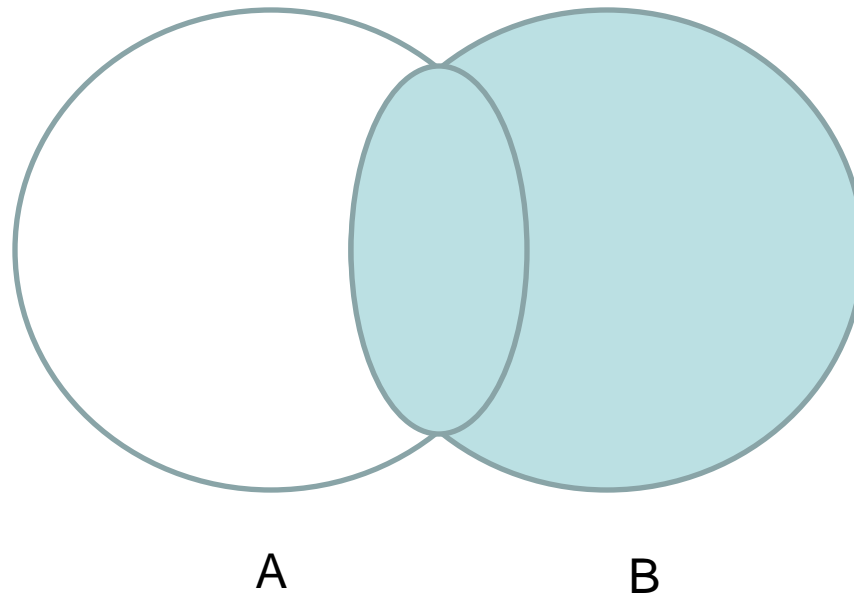


Table Joins: OUTER JOIN

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

```
7 rows in set (0.00 sec)
```

```
mysql> DESCRIBE PathwayGene;
```

Field	Type	Null	Key	Default	Extra
Name	varchar(50)	NO	PRI	NULL	
Gene	varchar(10)	NO	PRI	NULL	

```
2 rows in set (0.00 sec)
```

```
mysql> SELECT DISTINCT p.Gene, g.GeneId  
-> FROM PathwayGene p LEFT OUTER JOIN Gene g ON (p.Gene=g.Name)
```

Table Joins: OUTER JOIN

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

```
7 rows in set (0.00 sec)
```

```
mysql> DESCRIBE PathwayGene;
```

Field	Type	Null	Key	Default	Extra
Name	varchar(50)	NO	PRI	NULL	
Gene	varchar(10)	NO	PRI	NULL	

```
2 rows in set (0.00 sec)
```

```
mysql> SELECT DISTINCT p.Gene, g.GeneId  
-> FROM PathwayGene p LEFT OUTER JOIN Gene g ON (p.Gene=g.Name)  
-> WHERE g.GeneId IS NULL;
```

Table Joins: OUTER JOIN

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> DESCRIBE PathwayGene;
```

Field	Type	Null	Key	Default	Extra
Name	varchar(50)	NO	PRI	NULL	
Gene	varchar(10)	NO	PRI	NULL	

2 rows in set (0.00 sec)

```
mysql> SELECT DISTINCT p.Gene, g.GeneId
-> FROM PathwayGene p LEFT OUTER JOIN Gene g ON (p.Gene=g.Name)
-> WHERE g.GeneId IS NULL;
```

Gene	GeneId
AGPAT6	NULL
ANKRD57	NULL
FAM73B	NULL
FAM82A2	NULL
MOSC2	NULL
OBFC2A	NULL
PTRF	NULL
SDPR	NULL
SQRDL	NULL
FXB	NULL

Aggregates: COUNT

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

```
7 rows in set (0.00 sec)
```

```
mysql> DESCRIBE PathwayGene;
```

Field	Type	Null	Key	Default	Extra
Name	varchar(50)	NO	PRI	NULL	
Gene	varchar(10)	NO	PRI	NULL	

```
2 rows in set (0.00 sec)
```

```
mysql> SELECT DISTINCT COUNT(p.Gene)
-> FROM PathwayGene p LEFT OUTER JOIN Gene g ON (p.Gene=g.Name)
-> WHERE g.GeneId IS NULL;
```

COUNT(p.Gene)
128

```
1 row in set (6.45 sec)
```

Aggregates: GROUP BY

```
mysql> DESCRIBE PathwayGene;
```

Field	Type	Null	Key	Default	Extra
Name	varchar(50)	NO	PRI	NULL	
Gene	varchar(10)	NO	PRI	NULL	

2 rows in set (0.00 sec)

```
mysql> SELECT Gene, COUNT(Name)  
-> FROM PathwayGene  
-> GROUP BY Gene  
-> LIMIT 10;
```

Aggregates: GROUP BY

```
mysql> DESCRIBE PathwayGene;
```

Field	Type	Null	Key	Default	Extra
Name	varchar(50)	NO	PRI	NULL	
Gene	varchar(10)	NO	PRI	NULL	

2 rows in set (0.00 sec)

```
mysql> SELECT Gene, COUNT(Name)  
-> FROM PathwayGene  
-> GROUP BY Gene  
-> LIMIT 10;
```

Gene	COUNT(Name)
A2M	2
AAAS	1
AADAT	1
AARS	1
ABAT	2
ABCA1	5
ABCA2	2
ABCA3	3
ABCA4	1
ABCA5	1

10 rows in set (0.02 sec)

Aggregates: ORDER BY

```
mysql> DESCRIBE PathwayGene;
```

Field	Type	Null	Key	Default	Extra
Name	varchar(50)	NO	PRI	NULL	
Gene	varchar(10)	NO	PRI	NULL	

2 rows in set (0.00 sec)

```
mysql> SELECT Gene, COUNT(Name)
-> FROM PathwayGene
-> GROUP BY Gene
-> ORDER BY COUNT(Name) DESC
-> LIMIT 10;
```

Gene	COUNT (Name)
MYC	10
CDKN1A	10
IL6	10
SERPINE1	9
RETSAT	8
PLAUR	8
CA2	8
IRF1	8
IDH1	8
BTG2	8

10 rows in set (0.01 sec)

Aggregates: HAVING

```
mysql> DESCRIBE PathwayGene;
```

Field	Type	Null	Key	Default	Extra
Name	varchar(50)	NO	PRI	NULL	
Gene	varchar(10)	NO	PRI	NULL	

2 rows in set (0.00 sec)

```
mysql> SELECT Gene, COUNT(Name)
-> FROM PathwayGene
-> GROUP BY Gene
-> HAVING COUNT(Name) > 9;
```

Gene	COUNT(Name)
CDKN1A	10
IL6	10
MYC	10

3 rows in set (0.01 sec)

MySQL Aggregate Functions

Table 12.25 Aggregate (GROUP BY) Functions

Name	Description
<u>AVG ()</u>	Return the average value of the argument
<u>BIT_AND ()</u>	Return bitwise AND
<u>BIT_OR ()</u>	Return bitwise OR
<u>BIT_XOR ()</u>	Return bitwise XOR
<u>COUNT ()</u>	Return a count of the number of rows returned
<u>COUNT (DISTINCT)</u>	Return the count of a number of different values
<u>GROUP_CONCAT ()</u>	Return a concatenated string
<u>MAX ()</u>	Return the maximum value
<u>MIN ()</u>	Return the minimum value
<u>STD ()</u>	Return the population standard deviation
<u>STDDEV ()</u>	Return the population standard deviation
<u>STDDEV_POP ()</u>	Return the population standard deviation
<u>STDDEV_SAMP ()</u>	Return the sample standard deviation
<u>SUM ()</u>	Return the sum
<u>VAR_POP ()</u>	Return the population standard variance
<u>VAR_SAMP ()</u>	Return the sample variance
<u>VARIANCE ()</u>	Return the population standard variance

<https://dev.mysql.com/doc/refman/5.7/en/group-by-functions.html>

SQL: Query Optimization

The best way to improve query efficiency is to index the columns in your query

- can be any number of columns
- can have multiple indexes for a single table

Without an index, entire table needs to be read

- table scan
- can be very time consuming on large tables

Primary Key Optimization

Primary key for a table

- column has an associated index created
- data is sorted and physically stored (clustered)
 - default storage engine (InnoDB)

*Common practice to create a separate auto-increment integer column to serve as the primary key if no obvious column exist

Primary Key Optimization

```
mysql> SELECT * FROM Gene LIMIT 10;
```

GeneId	Name	Biotype	Chr	Start	End	Strand
ENSG00000223972	DDX11L1	transcribed_unprocessed_pseudogene	1	11869	14409	+
ENSG00000227232	WASH7P	unprocessed_pseudogene	1	14404	29570	-
ENSG00000278267	MIR6859-1	miRNA	1	17369	17436	-
ENSG00000243485	MIR1302-2HG	lincRNA	1	29554	31109	+
ENSG00000284332	MIR1302-2	miRNA	1	30366	30503	+
ENSG00000237613	FAM138A	lincRNA	1	34554	36081	-
ENSG00000268020	OR4G4P	unprocessed_pseudogene	1	52473	53312	+
ENSG00000240361	OR4G11P	transcribed_unprocessed_pseudogene	1	57598	64116	+
ENSG00000186092	OR4F5	protein_coding	1	65419	71585	+
ENSG00000238009	AL627309.1	lincRNA	1	89295	133723	-

10 rows in set (0.00 sec)

```
mysql> SELECT * FROM Gene LIMIT 10;
```

GeneId	Name	Biotype	Chr	Start	End	Strand
ENSG000000000003	TSPAN6	protein_coding	X	100627109	100639991	-
ENSG000000000005	TNMD	protein_coding	X	100584802	100599885	+
ENSG000000000419	DPM1	protein_coding	20	50934867	50958555	-
ENSG000000000457	SCYL3	protein_coding	1	169849631	169894267	-
ENSG000000000460	Clorf112	protein_coding	1	169662007	169854080	+
ENSG000000000938	FGR	protein_coding	1	27612064	27635277	-
ENSG000000000971	CFH	protein_coding	1	196651878	196747504	+
ENSG00000001036	FUCA2	protein_coding	6	143494811	143511690	-
ENSG00000001084	GCLC	protein_coding	6	53497341	53616970	-
ENSG00000001167	NFYA	protein_coding	6	41072945	41099976	+

10 rows in set (0.00 sec)

MySQL Database Indexes

Cluster index

- primary key

Secondary indexes

- indexed columns based on specific queries
- references the primary key (InnoDB)
- better to have shorter primary keys
 - storage considerations

Aggregates: COUNT

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

```
7 rows in set (0.00 sec)
```

```
mysql> DESCRIBE PathwayGene;
```

Field	Type	Null	Key	Default	Extra
Name	varchar(50)	NO	PRI	NULL	
Gene	varchar(10)	NO	PRI	NULL	

```
2 rows in set (0.00 sec)
```

```
mysql> SELECT DISTINCT COUNT(p.Gene)
-> FROM PathwayGene p LEFT OUTER JOIN Gene g ON (p.Gene=g.Name)
-> WHERE g.GeneId IS NULL;
```

COUNT(p.Gene)
128

```
1 row in set (6.45 sec)
```


SQL: EXPLAIN

```
mysql> SELECT DISTINCT COUNT(p.Gene)
      -> FROM PathwayGene p LEFT OUTER JOIN Gene g ON (p.Gene=g.Name)
      -> WHERE g.GeneId IS NULL;
+-----+
| COUNT(p.Gene) |
+-----+
|          128 |
+-----+
1 row in set (6.45 sec)
```

```
mysql> EXPLAIN SELECT DISTINCT COUNT(p.Gene)
      -> FROM PathwayGene p LEFT OUTER JOIN Gene g ON (p.Gene=g.Name)
      -> WHERE g.GeneId IS NULL;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | p | NULL | index | NULL | PRIMARY | 64 | NULL | 7316 | 100.00 | Using index |
| 1 | SIMPLE | g | NULL | ALL | NULL | NULL | NULL | NULL | 61482 | 10.00 | Using where; Not exists; Using join buffer (Block Nested Loop) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)
```

SQL: EXPLAIN

Table 8.1 EXPLAIN Output Columns

Column	JSON Name	Meaning
<u>id</u>	select_id	The <code>SELECT</code> identifier
<u>select_type</u>	None	The <code>SELECT</code> type
<u>table</u>	table_name	The table for the output row
<u>partitions</u>	partitions	The matching partitions
<u>type</u>	access_type	The join type
<u>possible_keys</u>	possible_keys	The possible indexes to choose
<u>key</u>	key	The index actually chosen
<u>key_len</u>	key_length	The length of the chosen key
<u>ref</u>	ref	The columns compared to the index
<u>rows</u>	rows	Estimate of rows to be examined
<u>filtered</u>	filtered	Percentage of rows filtered by table condition
<u>Extra</u>	None	Additional information

SQL: CREATE INDEX

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> DESCRIBE PathwayGene;
```

Field	Type	Null	Key	Default	Extra
Name	varchar(50)	NO	PRI	NULL	
Gene	varchar(10)	NO	PRI	NULL	

2 rows in set (0.00 sec)

```
mysql> CREATE INDEX name ON Gene (Name);
```

Query OK, 0 rows affected (0.13 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
mysql> EXPLAIN SELECT DISTINCT COUNT(p.Gene)
```

```
-> FROM PathwayGene p LEFT OUTER JOIN Gene g ON (p.Gene=g.Name)
```

```
-> WHERE g.GeneId IS NULL;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	p	NULL	index	NULL	PRIMARY	64	NULL	7316	100.00	Using index
1	SIMPLE	g	NULL	ref	name	name	22	zheng.p.Gene	1	10.00	Using where; Not exists; Using index

2 rows in set, 1 warning (0.00 sec)

SQL: INDEX

```
mysql> EXPLAIN SELECT DISTINCT COUNT(p.Gene)
-> FROM PathwayGene p LEFT OUTER JOIN Gene g ON (p.Gene=g.Name)
-> WHERE g.GeneId IS NULL;
+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+
| 1 | SIMPLE | p | NULL | index | NULL | PRIMARY | 64 | NULL | 7316 | 100.00 | Using index |
| 1 | SIMPLE | g | NULL | ALL | NULL | NULL | NULL | NULL | 61482 | 10.00 | Using where; Not exists; Using join buffer (Block Nested Loop) |
+-----+
2 rows in set, 1 warning (0.00 sec)
```

```
mysql> CREATE INDEX name ON Gene (Name);
Query OK, 0 rows affected (0.13 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> EXPLAIN SELECT DISTINCT COUNT(p.Gene)
-> FROM PathwayGene p LEFT OUTER JOIN Gene g ON (p.Gene=g.Name)
-> WHERE g.GeneId IS NULL;
+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+
| 1 | SIMPLE | p | NULL | index | NULL | PRIMARY | 64 | NULL | 7316 | 100.00 | Using index |
| 1 | SIMPLE | g | NULL | ref | name | name | 22 | zheng.p.Gene | 1 | 10.00 | Using where; Not exists; Using index |
+-----+
2 rows in set, 1 warning (0.00 sec)
```

SQL: Query Optimization

```
mysql> SELECT DISTINCT COUNT(p.Gene)
-> FROM PathwayGene p LEFT OUTER JOIN Gene g ON (p.Gene=g.Name)
-> WHERE g.GeneId IS NULL;
+-----+
| COUNT(p.Gene) |
+-----+
|          128 |
+-----+
1 row in set (6.45 sec)
```

```
mysql> SELECT DISTINCT COUNT(p.Gene)
-> FROM PathwayGene p LEFT OUTER JOIN Gene g ON (p.Gene=g.Name)
-> WHERE g.GeneId IS NULL;
+-----+
| COUNT(p.Gene) |
+-----+
|          128 |
+-----+
1 row in set (0.04 sec)
```

SQL Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

List all genes on Chr11 between
positions 55786964 and 135075899?

SQL Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> SELECT Name, Chr, Start, End, Strand
-> FROM Gene
-> WHERE Chr=11 AND Start >= 55786964 AND End <= 135075899
-> LIMIT 10;
```

Name	Chr	Start	End	Strand
BAD	11	64269830	64284704	-
TMEM132A	11	60924463	60937159	+
ALDH3B1	11	68008578	68029282	+
SYT7	11	61515313	61581148	-
CD6	11	60971680	61020377	+
POLA2	11	65261762	65305589	+
CAPN1	11	65180566	65212006	+
VSIG2	11	124747472	124752238	-
PHLDB1	11	118606440	118658038	+
MRE11	11	94415578	94493908	-

SQL Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

List all overlapping genes?

SQL Query

```
mysql> DESCRIBE Gene;
```

Field	Type	Null	Key	Default	Extra
GeneId	char(15)	NO		NULL	
Name	varchar(20)	NO		NULL	
Biotype	varchar(50)	YES		NULL	
Chr	char(2)	YES		NULL	
Start	int(11)	YES		NULL	
End	int(11)	YES		NULL	
Strand	enum('-', '+')	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> SELECT g1.Name as Name1, g1.Chr as Chr1, g1.Start as Start1, g1.End as End1, g1.Strand as Strand1,  
->      g2.Name as Name2, g2.Chr as Chr2, g2.Start as Start2, g2.End as End2, g2.Strand as Strand2  
-> FROM Gene g1, Gene g2  
-> WHERE g1.Chr=g2.Chr AND (g1.Start <= g2.End AND g1.End >= g2.Start) AND g1.Name != g2.Name;
```

SELF JOIN

Saving a Query in a Table

```
mysql> CREATE TABLE OverlappingGenes (  
->     Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
->     Name1 VARCHAR(20) NOT NULL,  
->     Chr1 CHAR(2),  
->     Start1 INT,  
->     End1 INT,  
->     Strand1 ENUM('-', '+'),  
->     Name2 VARCHAR(20) NOT NULL,  
->     Chr2 CHAR(2),  
->     Start2 INT,  
->     End2 INT,  
->     Strand2 ENUM('-', '+')  
-> )  
-> AS  
-> SELECT g1.Name as Name1, g1.Chr as Chr1, g1.Start as Start1, g1.End as End1, g1.Strand as Strand1,  
->        g2.Name as Name2, g2.Chr as Chr2, g2.Start as Start2, g2.End as End2, g2.Strand as Strand2  
-> FROM Gene g1, Gene g2  
-> WHERE g1.Chr=g2.Chr AND (g1.Start <= g2.End AND g1.End >= g2.Start) AND g1.Name != g2.Name;  
Query OK, 56788 rows affected (5 min 53.70 sec)  
Records: 56788  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM OverlappingGenes LIMIT 10;
```

Id	Name1	Chr1	Start1	End1	Strand1	Name2	Chr2	Start2	End2	Strand2
1	C1orf112	1	169662007	169854080	+	SCYL3	1	169849631	169894267	-
2	SCYL3	1	169849631	169894267	-	C1orf112	1	169662007	169854080	+
3	SELE	1	169722641	169764705	-	C1orf112	1	169662007	169854080	+
4	METTL18	1	169792529	169794966	-	C1orf112	1	169662007	169854080	+
5	OARD1	6	41033627	41097787	-	NFYA	6	41072945	41099976	+
6	NIPAL3	1	24415794	24472976	+	STPG1	1	24356999	24416934	-
7	GRHL3	1	24319322	24364482	+	STPG1	1	24356999	24416934	-
8	STPG1	1	24356999	24416934	-	NIPAL3	1	24415794	24472976	+
9	CTTNBP2	7	117710651	117874139	-	CFTR	7	117465784	117715971	+
10	AC000061	7	117604791	117647415	-	CFTR	7	117465784	117715971	+

Saving a Query in a Table

```
mysql> CREATE TABLE OverlappingGenes (  
->     Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
->     Name1 VARCHAR(20) NOT NULL,  
->     Chr1 CHAR(2),  
->     Start1 INT,  
->     End1 INT,  
->     Strand1 ENUM('-', '+'),  
->     Name2 VARCHAR(20) NOT NULL,  
->     Chr2 CHAR(2),  
->     Start2 INT,  
->     End2 INT,  
->     Strand2 ENUM('-', '+')  
-> )  
-> AS  
-> SELECT g1.Name as Name1, g1.Chr as Chr1, g1.Start as Start1, g1.End as End1, g1.Strand as Strand1,  
->        g2.Name as Name2, g2.Chr as Chr2, g2.Start as Start2, g2.End as End2, g2.Strand as Strand2  
-> FROM Gene g1, Gene g2  
-> WHERE g1.Chr=g2.Chr AND (g1.Start <= g2.End AND g1.End >= g2.Start) AND g1.Name != g2.Name;  
Query OK, 56788 rows affected (5 min 53.70 sec)  
Records: 56788  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM OverlappingGenes LIMIT 10;
```

Id	Name1	Chr1	Start1	End1	Strand1	Name2	Chr2	Start2	End2	Strand2
1	C1orf112	1	169662007	169854080	+	SCYL3	1	169849631	169894267	-
2	SCYL3	1	169849631	169894267	-	C1orf112	1	169662007	169854080	+
3	SELE	1	169722641	169764705	-	C1orf112	1	169662007	169854080	+
4	METTL18	1	169792529	169794966	-	C1orf112	1	169662007	169854080	+
5	OARD1	6	41033627	41097787	-	NFYA	6	41072945	41099976	+
6	NIPAL3	1	24415794	24472976	+	STPG1	1	24356999	24416934	-
7	GRHL3	1	24319322	24364482	+	STPG1	1	24356999	24416934	-
8	STPG1	1	24356999	24416934	-	NIPAL3	1	24415794	24472976	+
9	CTTNBP2	7	117710651	117874139	-	CFTR	7	117465784	117715971	+
10	AC000061	7	117604791	117647415	-	CFTR	7	117465784	117715971	+

SQL Query

```
mysql> CREATE TABLE OverlappingGenes (
->
->     Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
->     Name1 VARCHAR(20) NOT NULL,
->     Chr1 CHAR(2),
->     Start1 INT,
->     End1 INT,
->     Strand1 ENUM('-', '+'),
->     Name2 VARCHAR(20) NOT NULL,
->     Chr2 CHAR(2),
->     Start2 INT,
->     End2 INT,
->     Strand2 ENUM('-', '+')
-> )
-> AS
-> SELECT g1.Name as Name1, g1.Chr as Chr1, g1.Start as Start1, g1.End as End1, g1.Strand as Strand1,
->        g2.Name as Name2, g2.Chr as Chr2, g2.Start as Start2, g2.End as End2, g2.Strand as Strand2
-> FROM Gene g1, Gene g2
-> WHERE g1.Chr=g2.Chr AND (g1.Start <= g2.End AND g1.End >= g2.Start) AND g1.Name != g2.Name;
Query OK, 56788 rows affected (5 min 53.70 sec)
Records: 56788  Duplicates: 0  Warnings: 0
```

Can we make this query more efficient/faster?

```
mysql> SELECT * FROM OverlappingGenes LIMIT 10;
```

Id	Name1	Chr1	Start1	End1	Strand1	Name2	Chr2	Start2	End2	Strand2
1	C1orf112	1	169662007	169854080	+	SCYL3	1	169849631	169894267	-
2	SCYL3	1	169849631	169894267	-	C1orf112	1	169662007	169854080	+
3	SELE	1	169722641	169764705	-	C1orf112	1	169662007	169854080	+
4	METTL18	1	169792529	169794966	-	C1orf112	1	169662007	169854080	+
5	OARD1	6	41033627	41097787	-	NFYA	6	41072945	41099976	+
6	NIPAL3	1	24415794	24472976	+	STPG1	1	24356999	24416934	-
7	GRHL3	1	24319322	24364482	+	STPG1	1	24356999	24416934	-
8	STPG1	1	24356999	24416934	-	NIPAL3	1	24415794	24472976	+
9	CTTNBP2	7	117710651	117874139	-	CFTR	7	117465784	117715971	+
10	AC000061	7	117604791	117647415	-	CFTR	7	117465784	117715971	+

SQL Query Optimization

```
mysql>
mysql> CREATE INDEX ChrStartEnd on Gene (Chr, Start, End);
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DROP TABLE OverlappingGenes;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE OverlappingGenes (
->
->     Id INT NOT NULL AUTO INCREMENT PRIMARY KEY,
->     Name1 VARCHAR(20) NOT NULL,
->     Chr1 CHAR(2),
->     Start1 INT,
->     End1 INT,
->     Strand1 ENUM('-', '+'),
->     Name2 VARCHAR(20) NOT NULL,
->     Chr2 CHAR(2),
->     Start2 INT,
->     End2 INT,
->     Strand2 ENUM('-', '+')
-> )
-> AS
-> SELECT g1.Name as Name1, g1.Chr as Chr1, g1.Start as Start1, g1.End as End1, g1.Strand as Strand1,
->        g2.Name as Name2, g2.Chr as Chr2, g2.Start as Start2, g2.End as End2, g2.Strand as Strand2
-> FROM Gene g1, Gene g2
-> WHERE g1.Chr=g2.Chr AND (g1.Start <= g2.End AND g1.End >= g2.Start) AND g1.Name != g2.Name;
Query OK, 56788 rows affected (41.71 sec)
Records: 56788 Duplicates: 0 Warnings: 0
```

SQL Query Optimization

```
mysql>
mysql> CREATE INDEX ChrStartEnd on Gene (Chr, Start, End);
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DROP TABLE OverlappingGenes;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE OverlappingGenes (
->
->     Id INT NOT NULL AUTO INCREMENT PRIMARY KEY,
->     Name1 VARCHAR(20) NOT NULL,
->     Chr1 CHAR(2),
->     Start1 INT,
->     End1 INT,
->     Strand1 ENUM('-', '+'),
->     Name2 VARCHAR(20) NOT NULL,
->     Chr2 CHAR(2),
->     Start2 INT,
->     End2 INT,
->     Strand2 ENUM('-', '+')
-> )
-> AS
-> SELECT g1.Name as Name1, g1.Chr as Chr1, g1.Start as Start1, g1.End as End1, g1.Strand as Strand1,
->        g2.Name as Name2, g2.Chr as Chr2, g2.Start as Start2, g2.End as End2, g2.Strand as Strand2
-> FROM Gene g1, Gene g2
-> WHERE g1.Chr=g2.Chr AND (g1.Start <= g2.End AND g1.End >= g2.Start) AND g1.Name != g2.Name;
Query OK, 56788 rows affected (41.71 sec)
Records: 56788 Duplicates: 0 Warnings: 0
```

SQL Query Optimization

```
mysql> EXPLAIN
-> SELECT g1.Name as Name1, g1.Chr as Chr1, g1.Start as Start1, g1.End as End1, g1.Strand as Strand1,
->         g2.Name as Name2, g2.Chr as Chr2, g2.Start as Start2, g2.End as End2, g2.Strand as Strand2
-> FROM Gene g1, Gene g2
-> WHERE g1.Chr=g2.Chr AND (g1.Start <= g2.End AND g1.End >= g2.Start) AND g1.Name != g2.Name;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	g1	NULL	ALL	NULL	NULL	NULL	NULL	58512	100.00	NULL
1	SIMPLE	g2	NULL	ALL	NULL	NULL	NULL	NULL	58512	1.00	Using where; Using join buffer (Block Nested Loop)

2 rows in set, 1 warning (0.00 sec)

```
mysql> CREATE INDEX ChrStartEnd on Gene (Chr, Start, End);
Query OK, 0 rows affected (0.15 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> EXPLAIN
-> SELECT g1.Name as Name1, g1.Chr as Chr1, g1.Start as Start1, g1.End as End1, g1.Strand as Strand1,
->         g2.Name as Name2, g2.Chr as Chr2, g2.Start as Start2, g2.End as End2, g2.Strand as Strand2
-> FROM Gene g1, Gene g2
-> WHERE g1.Chr=g2.Chr AND (g1.Start <= g2.End AND g1.End >= g2.Start) AND g1.Name != g2.Name;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	g1	NULL	ALL	ChrStartEnd	NULL	NULL	NULL	58512	100.00	Using where
1	SIMPLE	g2	NULL	ref	ChrStartEnd	ChrStartEnd	3	zheng.g1.Chr	2340	10.00	Using index condition; Using where

2 rows in set, 1 warning (0.00 sec)

Bioinformatics Databases

The image displays a collage of three major bioinformatics databases: NCBI, Ensembl, and the UCSC Genome Browser Gateway.

NCBI (National Center for Biotechnology Information): The top left interface shows the NCBI homepage with a navigation menu on the left. The menu includes links to 'All Databases', 'Assembly', 'BioCollections', 'BioProject', 'BioSample', 'BioSystems', 'Books', 'ClinVar', 'Clone', 'Conserved Domains', 'dbGaP', 'dbVar', 'EST', 'Gene', 'Genome', 'GEO DataSets', 'GEO Profiles', 'GSS', 'GTR', 'HomoloGene', 'Proteins', 'Sequence Analysis', 'Taxonomy', 'Training & Tutorials', and 'Variation'. The main content area features sections for 'Submit', 'Download', 'Learn', 'Develop', 'Analyze', and 'Research'.

Ensembl: The top right interface shows the Ensembl genome browser. It features a search bar with a dropdown menu for 'All species' listing various organisms like Chinese hamster, Chinese hamster, Chinese softshell turtle, Cod, Coelacanth, Cow, Darnara mole rat, Degu, Dog, Dolphin, Duck, Elephant, Ferret, Flycatcher, Fruitfly, Fugu, Gibbon, Golden Hamster, Gorilla, and Guinea Pig. The main content area includes sections for 'Browse a Genome', 'Favourite genomes', 'Variant Effect Predictor', 'Gene expression in different tissues', 'Retrieve gene sequence', 'Compare genes across species', 'Find SNPs and other variants for my gene', and 'Use my own data in Ensembl'.

UCSC Genome Browser Gateway: The bottom interface shows the UCSC Genome Browser Gateway. It features a 'Browse/Select Species' section with a tree view of various species. The main content area includes a 'Find Position' section with a search bar and a 'Human Genes Browser' section with a list of genes and their coordinates. The bottom section includes 'Download sequence and annotation data' and 'Assembly Details'.

Database Dumps



hgdownload.soe.ucsc.edu/goldenPath/hg38/database/

omBlastTab.txt.gz	03-Jul-2016 06:58	154K
drBlastTab.sql	10-Sep-2017 17:52	1.7K
drBlastTab.txt.gz	10-Sep-2017 17:52	346K
ensemblLift.sql	16-Apr-2017 22:27	1.3K
ensemblLift.txt.gz	16-Apr-2017 22:27	2.1K
estOrientInfo.sql	09-Jul-2017 10:43	1.8K
estOrientInfo.txt.gz	09-Jul-2017 10:43	105M
extFile.sql	30-Oct-2016 08:16	1.4K
extFile.txt.gz	30-Oct-2016 08:16	15K
extHg38Patch11.sql	10-Sep-2017 17:57	1.4K
extHg38Patch11.txt.gz	10-Sep-2017 17:57	1.7K
extNcbiAltSequence.sql	06-Mar-2014 16:49	1.4K
extNcbiAltSequence.txt.gz	06-Mar-2014 16:49	3.3K
extNcbiRefSeq.sql	06-Mar-2017 03:56	1.4K
extNcbiRefSeq.txt.gz	06-Mar-2017 03:56	88
fishClones.sql	06-Mar-2016 06:27	1.9K
fishClones.txt.gz	06-Mar-2016 06:27	273K
foldUtr3.sql	03-Jul-2016 07:07	1.4K
foldUtr3.txt.gz	03-Jul-2016 07:07	30M
foldUtr5.sql	03-Jul-2016 07:08	1.4K
foldUtr5.txt.gz	03-Jul-2016 07:08	8.7M
gap.sql	06-Mar-2014 16:53	1.6K
gap.txt.gz	06-Mar-2014 16:53	12K
gbLoaded.sql	31-Dec-2017 08:00	1.6K
gbLoaded.txt.gz	31-Dec-2017 08:00	8.2K
gc5BaseBw.sql	06-Mar-2014 16:47	1.3K
gc5BaseBw.txt.gz	06-Mar-2014 16:47	63
geneReviews.sql	31-Dec-2017 08:07	1.5K
geneReviews.txt.gz	31-Dec-2017 08:07	27K
geneReviewsDetail.sql	31-Dec-2017 08:08	1.4K
geneReviewsDetail.txt.gz	31-Dec-2017 08:08	26K
geneid.sql	22-Nov-2015 18:41	1.9K
geneid.txt.gz	22-Nov-2015 18:41	2.3M
genomicSuperDups.sql	19-Oct-2014 12:48	2.4K
genomicSuperDups.txt.gz	19-Oct-2014 12:48	4.0M
genscan.sql	06-Mar-2014 16:48	1.7K
genscan.txt.gz	06-Mar-2014 16:48	3.1M
gnfAtlas2.sql	11-May-2015 01:50	1.9K

Database Dumps

```
zheng@state:~/BMI535635/gene$ mysqldump -u zheng -p --databases zheng > zheng.sql
Enter password:
```

0

```
-- Table structure for table `Exon`
--
```

```
DROP TABLE IF EXISTS `Exon`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Exon` (
  `ExonId` char(15) NOT NULL,
  `Chr` char(2) DEFAULT NULL,
  `Start` int(11) DEFAULT NULL,
  `End` int(11) DEFAULT NULL,
  `Strand` enum('-', '+') DEFAULT NULL,
  PRIMARY KEY (`ExonId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character set client = @saved cs client */;
```

```
--
-- Dumping data for table `Exon`
--
```

```
LOCK TABLES `Exon` WRITE;
/*!40000 ALTER TABLE `Exon` DISABLE KEYS */;
INSERT INTO `Exon` VALUES ('ENSE00000327880','1',27732603,27732657,'+'),('ENSE00000328922','417,20243110','-'),('ENSE00000331106','22',19148576,19149095','-'),('ENSE00000331191','22',195000331859','-'),('ENSE00000331693','11',2412754,2413012','-'),('ENSE00000331695','11',2411635,2411767,-0000331859','11',14464911,14465030,-'),('ENSE00000332771','X',48688054,48688096,+'),('ENSE00000332771','8',132897649,132897786,+'),('ENSE00000333729','8',132898798,132898910,+'),('ENSE00000333729','08009759,108009887,+'),('ENSE00000334741','15',34915038,34915179,-'),('ENSE00000334743','15,76738993,+'),('ENSE00000335892','16',56869729,56869824,+'),('ENSE00000335893','16',568700000336027','16',56892953,56893054,+'),('ENSE00000336287','19',1632332,1632405E00000339040','11',17520870,17520994,-'),('ENSE00000339472','4',99853218,99853348,+'),('ENSE00000341216','15',29108270,29108389,+'),('ENSE00000341216','19',14160112,14160297,-'),('ENSE00000341216','07986,53508222,-'),('ENSE00000341649','19',39707178,39707388,+'),('ENSE00000341661','19',39707178,39707388,+);
```

Database Dumps

Exporting


```
zheng@state:~/BMI535635/gene$ mysqldump -u zheng -p --databases zheng Pathway > zheng.Pathway.sql  
Enter password:
```

Importing

```
zheng@state:~/BMI535635/gene$ mysql -u zheng -p zheng < zheng.Pathway.sql  
Enter password:
```

UCSC Genome Browser

Securehttps://genome.ucsc.edu/goldenpath/help/mysql.html

GenomesGenome BrowserToolsMirrorsDownloadsMy DataHelpAbout Us

Downloading Data using MySQL

We have two MySQL databases for public access:

- **genome-mysql.soe.ucsc.edu** (located on the US west coast)
- **genome-euro-mysql.soe.ucsc.edu** (located in Europe)

These servers allow MySQL access to the same set of data currently available on our public Genome Browser site. The data are synchronized weekly with the main databases on our public site. During synchronization, the MySQL server can be intermittently out of sync with the main website for a short period of time. The weekly synchronization takes place on Monday mornings from 4:00 am to 9:00 am Pacific Time (GMT -7:00).

Connecting

You must have MySQL client libraries installed on your computer. We recommend using the most current version of the [v5.0 MySQL clients](#).

You can connect to the US MySQL server using the command:

```
mysql --user=genome --host=genome-mysql.soe.ucsc.edu -A
```

Or the European MySQL server with this command:

```
mysql --user=genome --host=genome-euro-mysql.soe.ucsc.edu -A
```

The -A flag is optional but is recommended for speed.

Once connected to the database, you may use a wide range of MySQL commands to query the database.

Conditions of use

- Avoid excessive or heavy queries that may impact the server performance. Inappropriate query use will result in a restriction of access. If you plan to execute a query that you think may be excessive, contact UCSC first to avoid the possibility of having your access blocked.
- Bot access and excessive program-driven use are not permitted.
- Attachments by local mirror sites are prohibited.

Using the MySQL server with our utilities

The MySQL database can also be used by the numerous utilities in the [Genome Browser source](#) tree. Some of these utilities require a password, so you will need to add the following specifications to your \$HOME/.hg.conf file (remember to chmod your .hg.conf file to 600 permissions) if you would like to access the US public MySQL server:

```
#US MySQL server
db.host=genome-mysql.soe.ucsc.edu
db.user=genomep
db.password=password
central.db=hgcentral
central.host=genome-mysql.soe.ucsc.edu
central.user=genomep
central.password=password
```