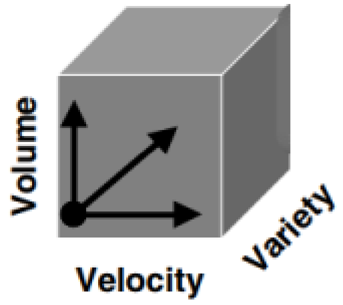
OREGON HEALTH & SCIENCE UNIVERSITY

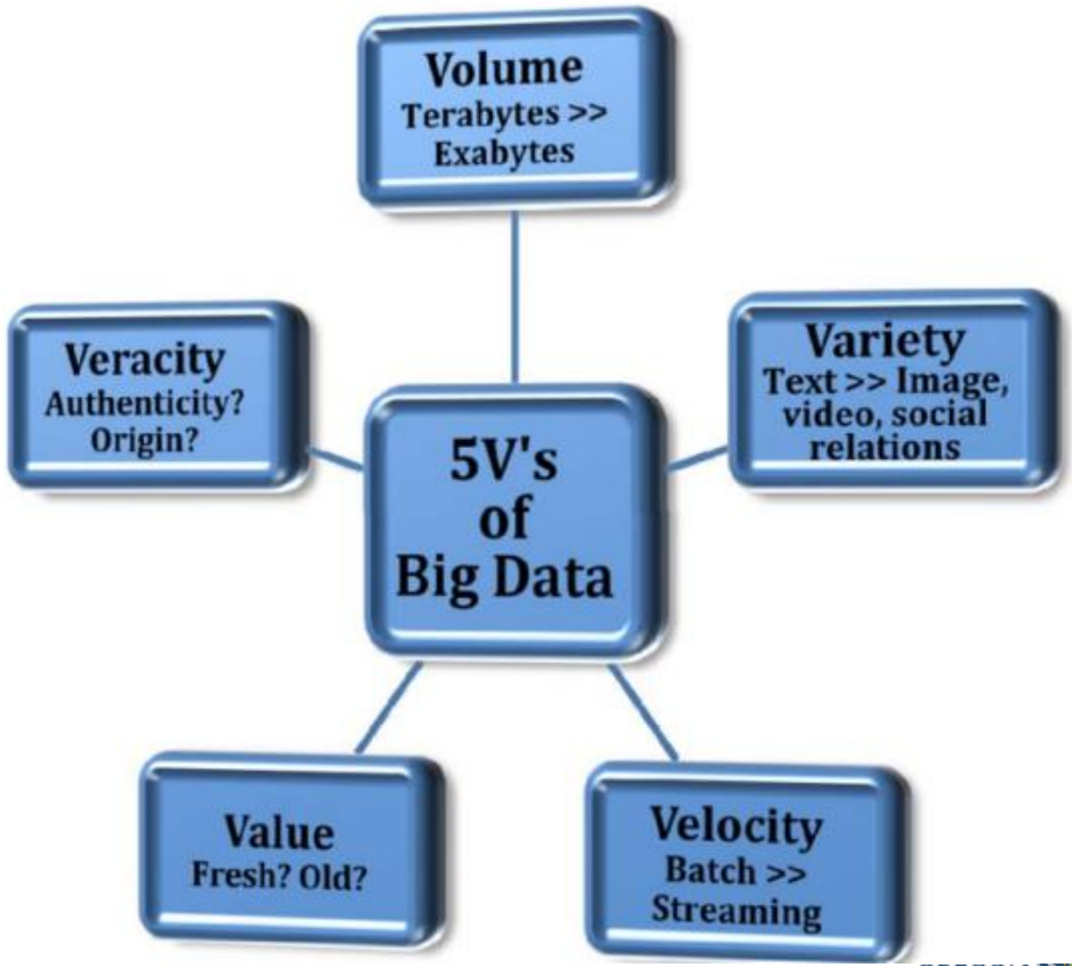
Processing and Management of Large Scale Data

BMI 535/635

Challenges of Big Data



Laney, META Group 2001



Yin and Kaynak, Proceedings in the IEEE 2015

Processing Big Data

Massively parallel processing

- coordinated processing by multiple processors working on different parts of a single program

Cluster Computing

- individual computer nodes performing tasks in a controlled and scheduled manner

Map-Reduce

- programming paradigm aimed at splitting one big processing job into many small ones (split-apply-combine)

OREGON HEALTH & SCIENCE UNIVERSITY

Storage/Querying

Relational Database Systems

- MySQL, Postgres, Oracle, Microsoft SQL

NoSQL

- document stores
    - CouchDB, MongoDB
- column stores
    - Hbase, BigTable
- graph databases
    - Neo4j, Dgraph

OREGON HEALTH & SCIENCE UNIVERSITY

1000 Genomes Project

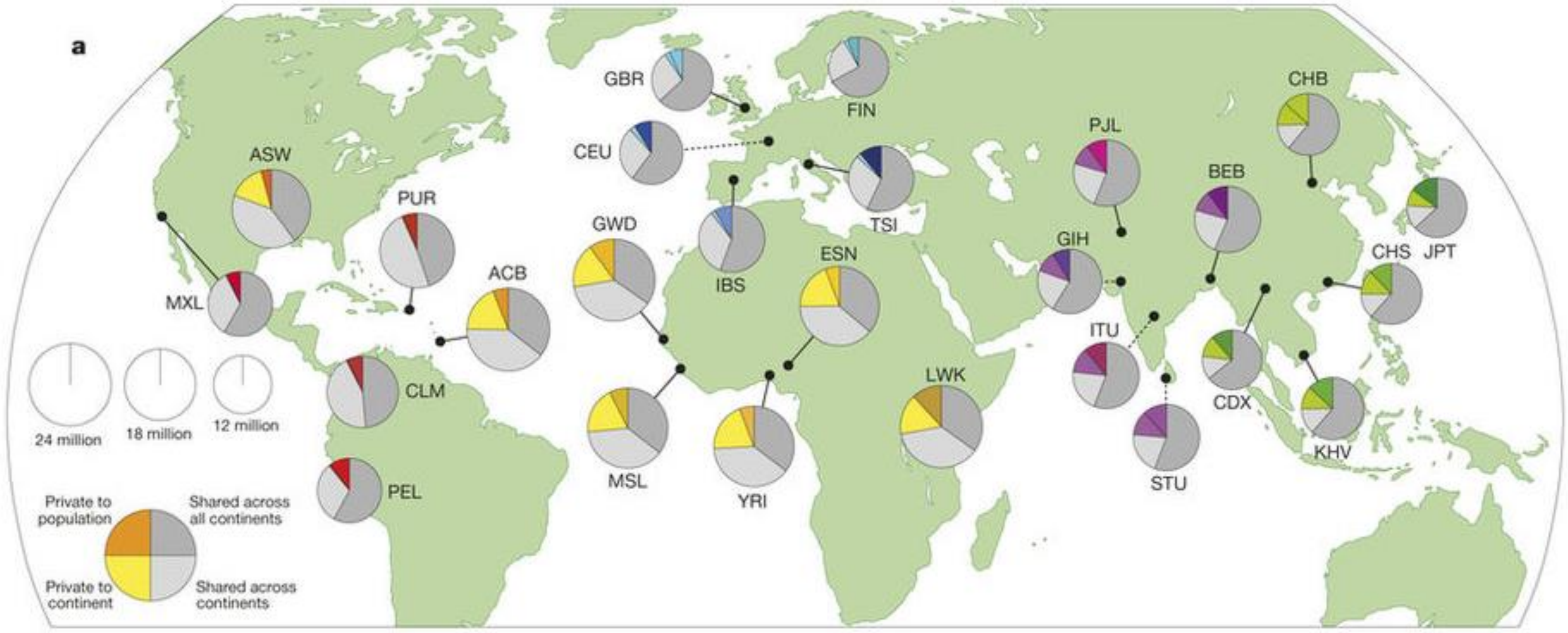
International project to construct a foundational data set of human genetics

- discover virtually all common human variation
    - single nucleotide polymorphisms
    - structural variants
    - haplotypes
- develop sequence analysis methods, tools, and reagents that can be transferred to other sequencing projects

OREGON HEALTH & SCIENCE UNIVERSITY

1000 Genomes Project

2504 individuals, 26 populations



The 1000 Genomes Project Consortium , Nature 2015

1000 Genomes Project

2504 individuals, 26 populations

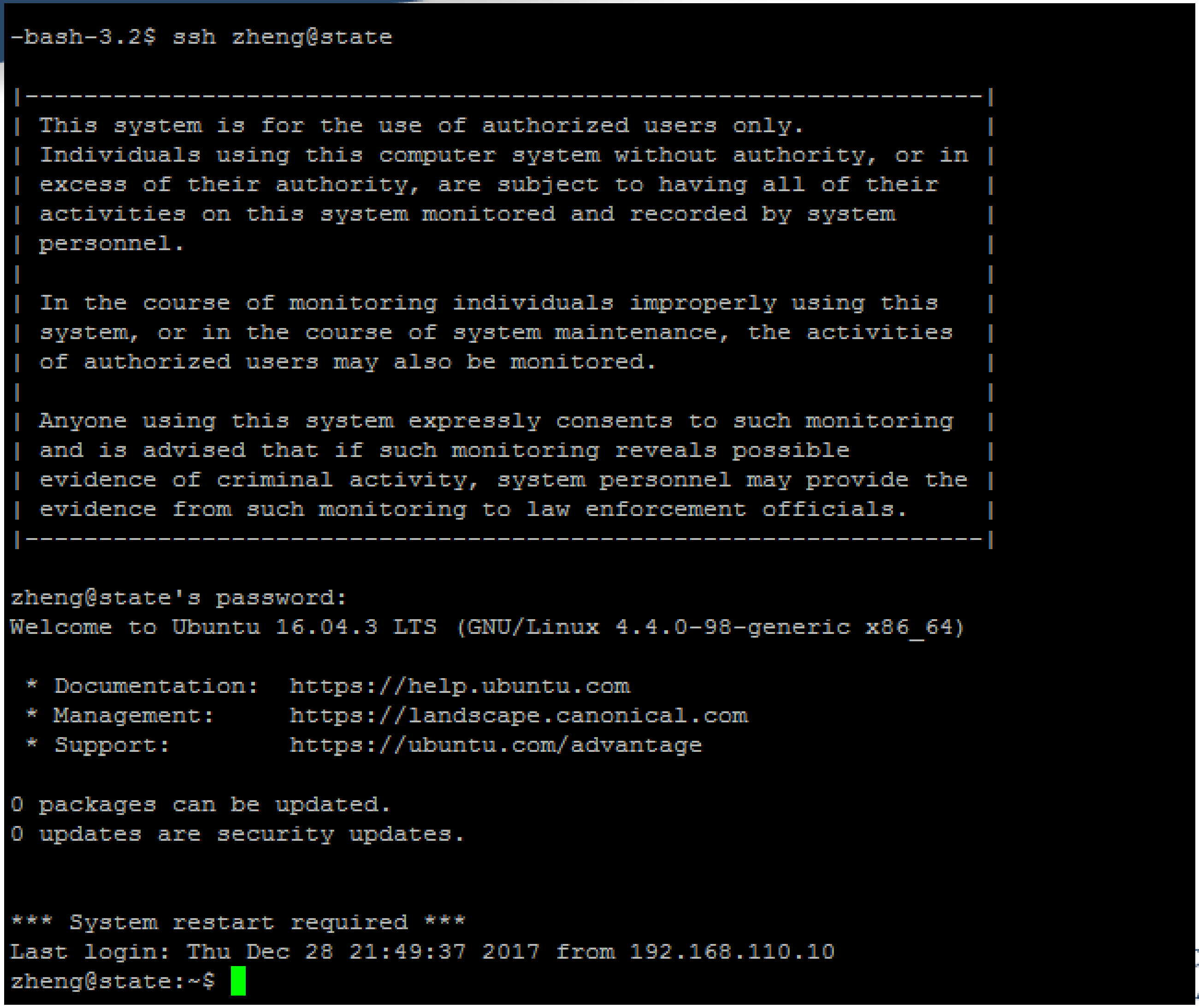
- low-coverage whole genome sequencing (mean of 7.4X)
- deep exome sequencing (mean of 65.7X)
- dense microarray genotyping

84.7M SNPs, 3.6 million indels, 60,000 structural variants

- >99% of SNP variants >1% frequency

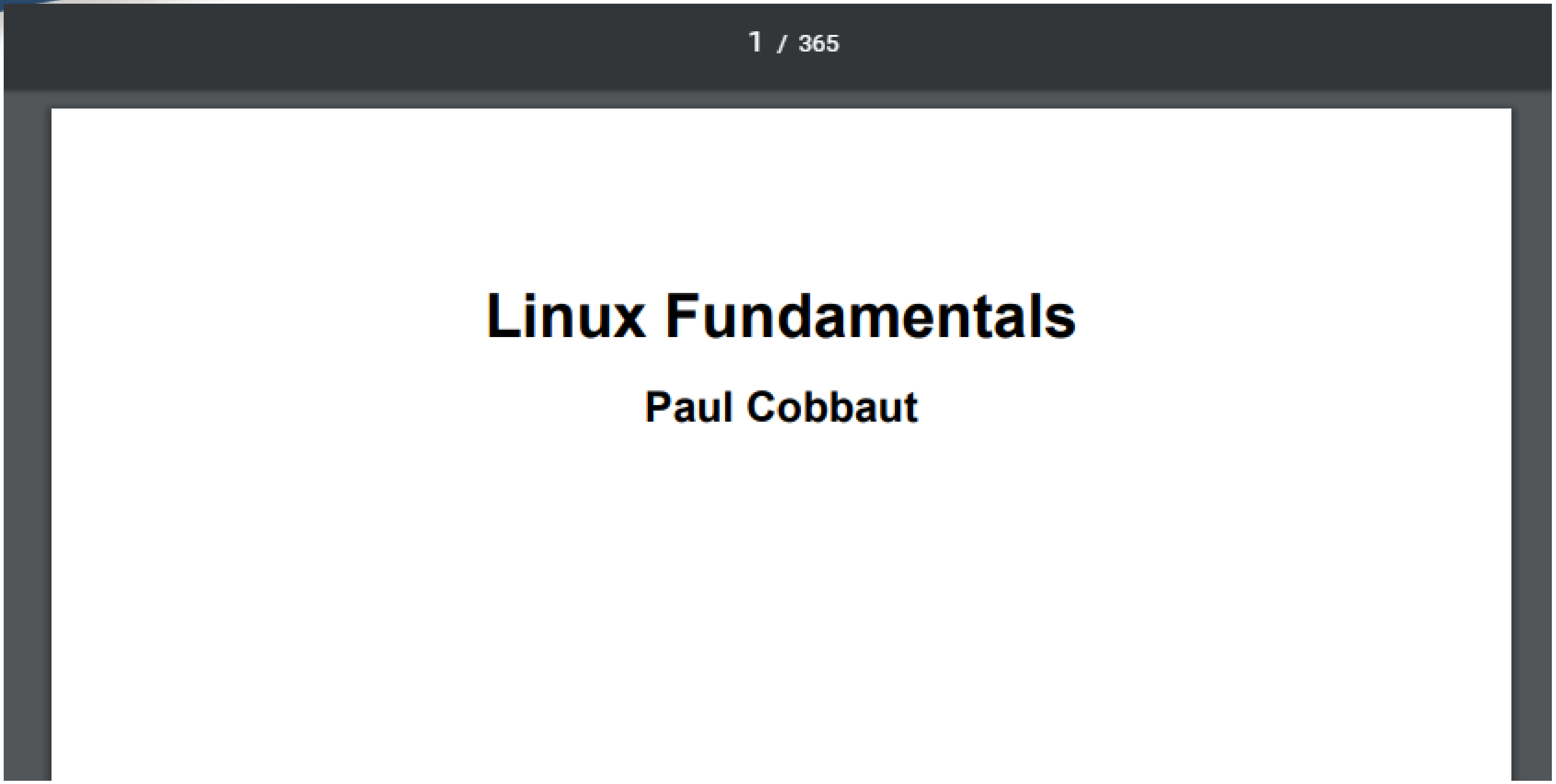
OREGON HEALTH & SCIENCE UNIVERSITY

State Server



```
-bash-3.2$ ssh zheng@state

|-----------------------------------------------------------------|
| This system is for the use of authorized users only.            |
| Individuals using this computer system without authority, or in |
| excess of their authority, are subject to having all of their   |
| activities on this system monitored and recorded by system      |
| personnel.                                                      |
|                                                                 |
| In the course of monitoring individuals improperly using this   |
| system, or in the course of system maintenance, the activities  |
| of authorized users may also be monitored.                      |
|                                                                 |
| Anyone using this system expressly consents to such monitoring  |
| and is advised that if such monitoring reveals possible         |
| evidence of criminal activity, system personnel may provide the |
| evidence from such monitoring to law enforcement officials.     |
|-----------------------------------------------------------------|

zheng@state's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-98-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Thu Dec 28 21:49:37 2017 from 192.168.110.10
zheng@state:~$
```

Linux Tutorial

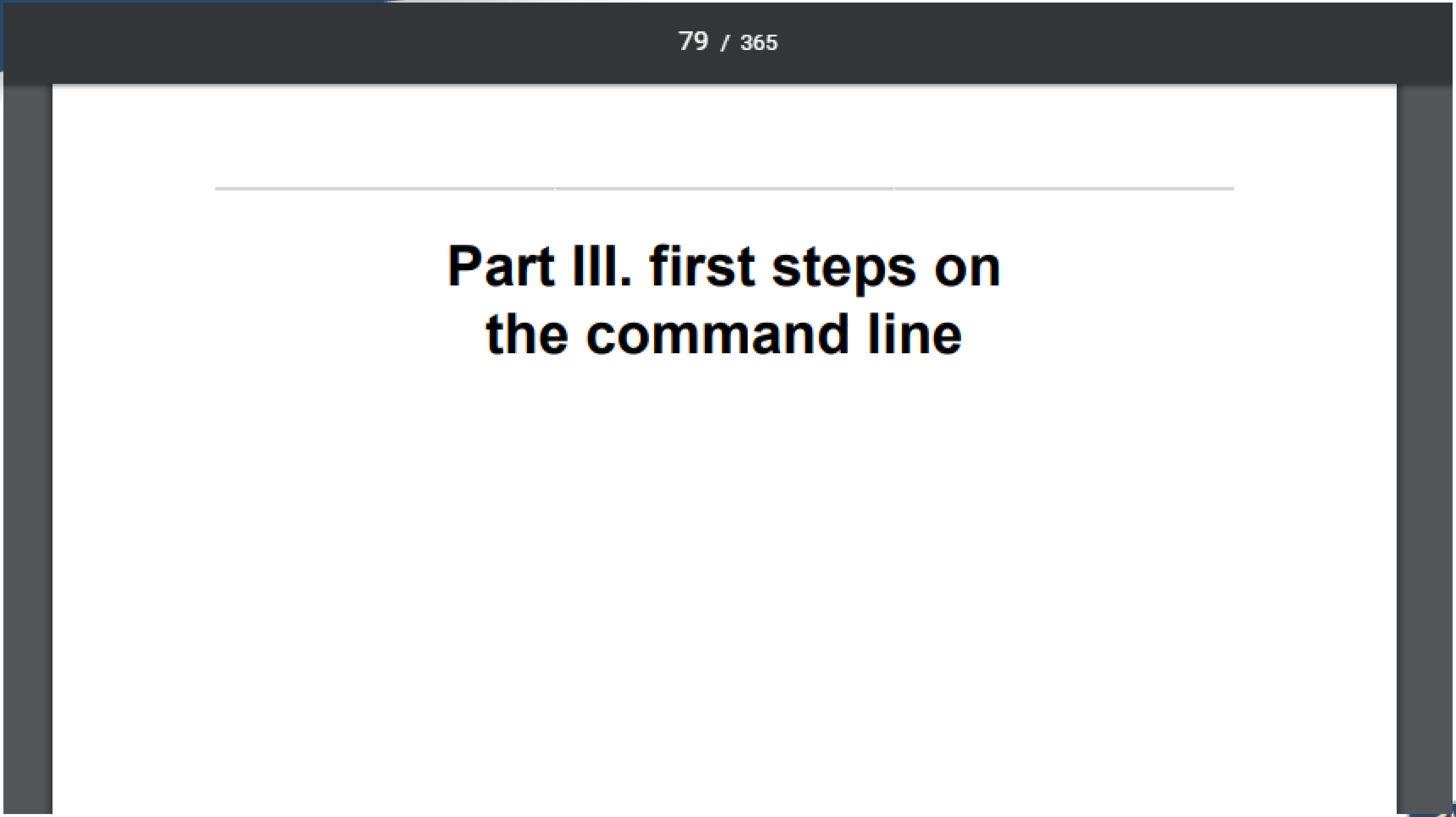


1 / 365

Linux Fundamentals

Paul Cobbaut

Linux Tutorial



79 / 365

Part III. first steps on the command line

Linux Tutorial

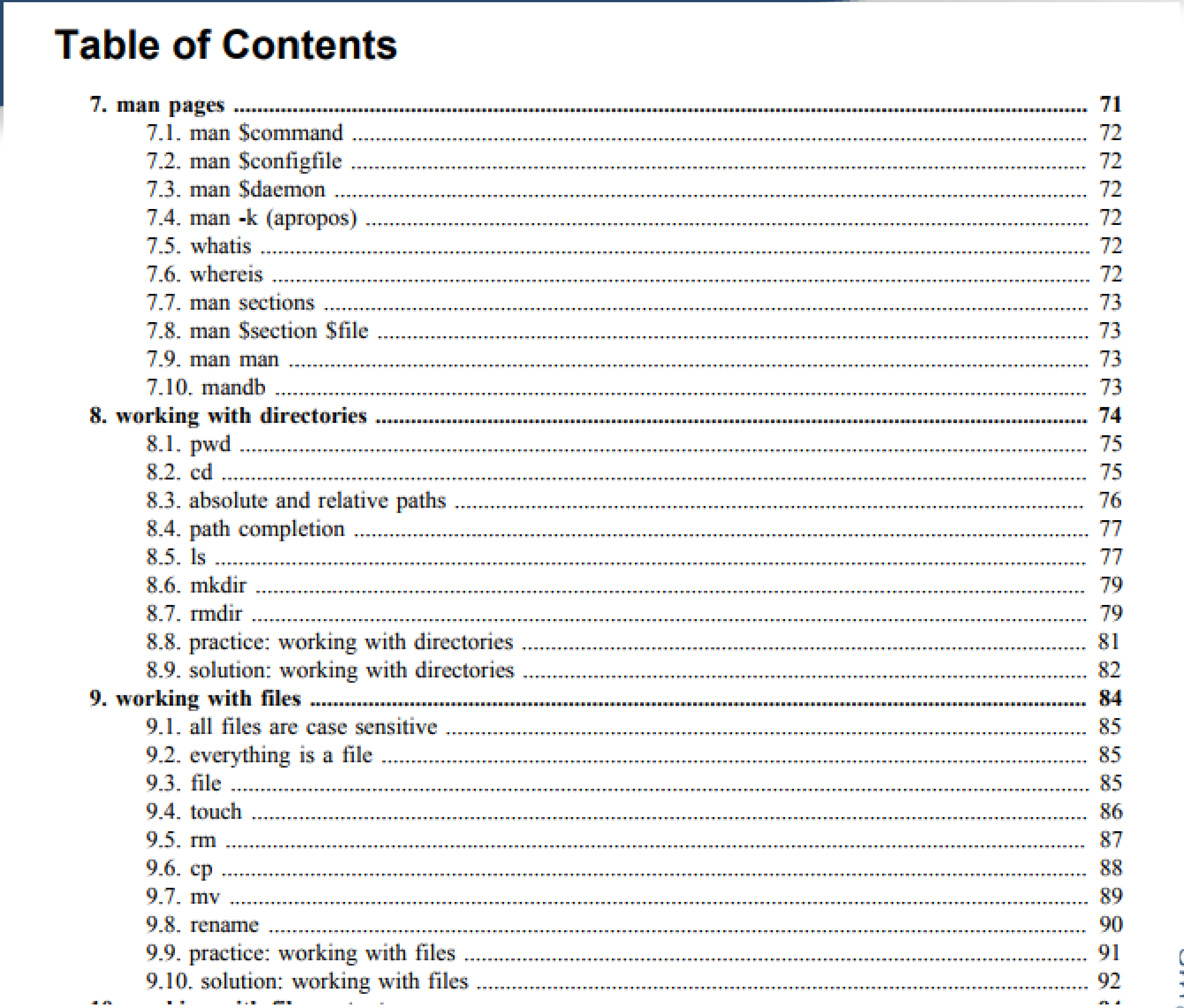


Table of Contents

Linux Tutorial

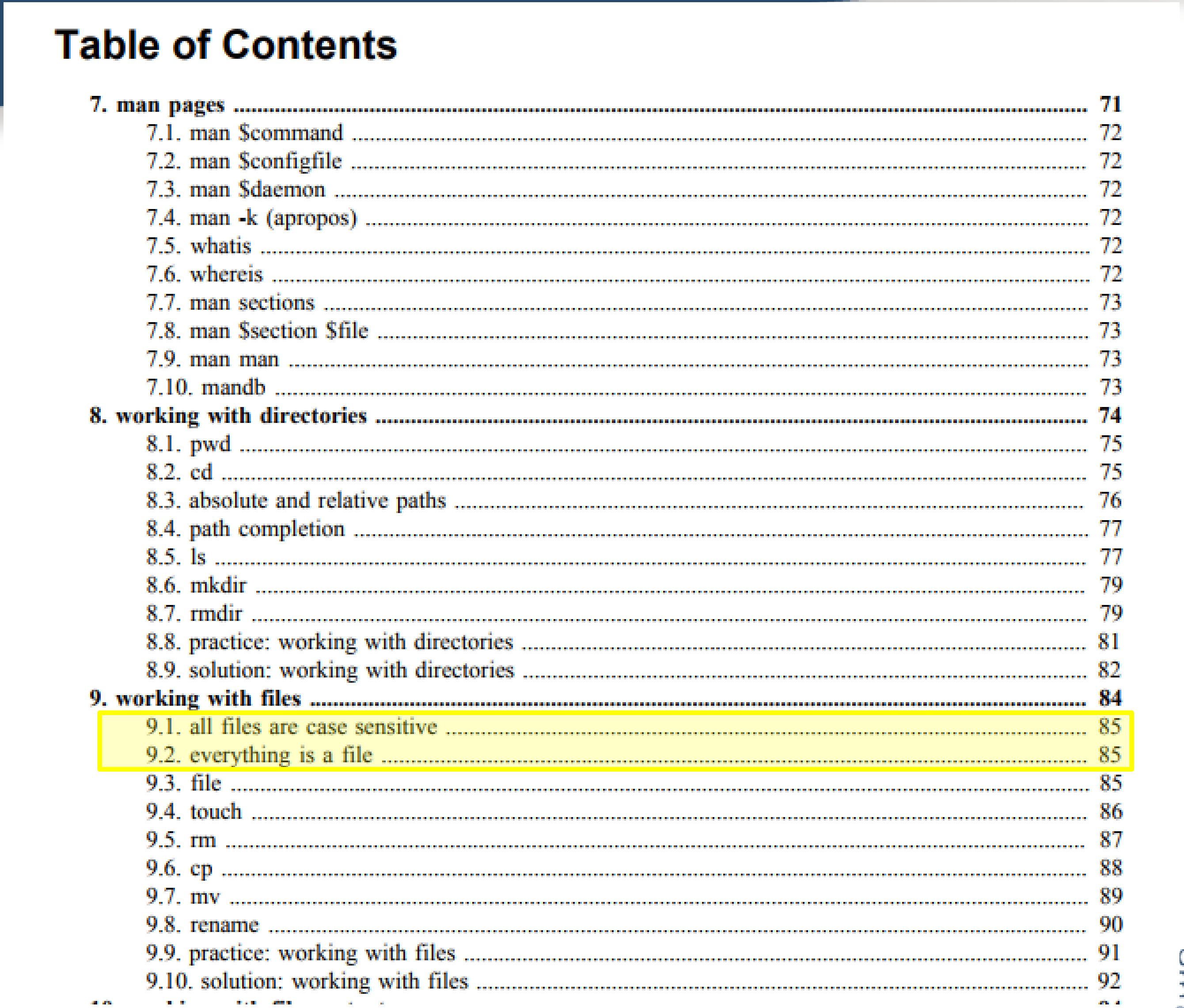
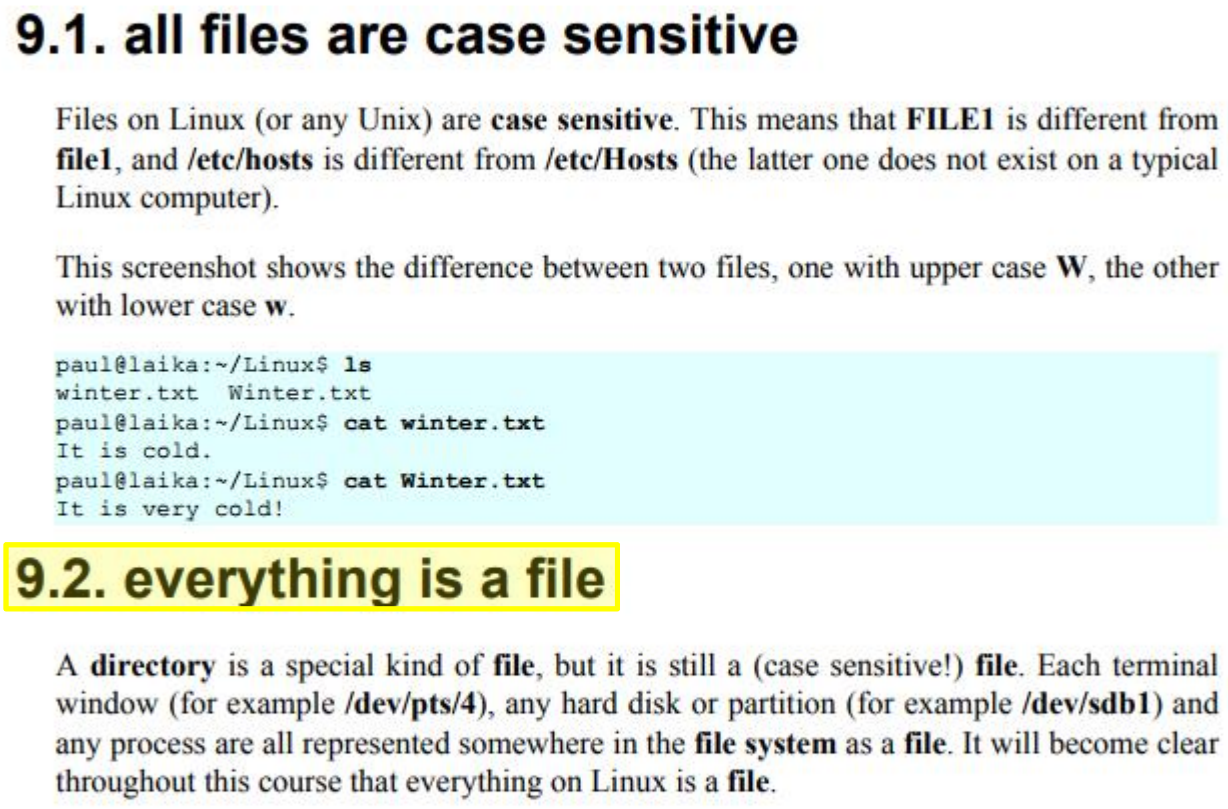


Table of Contents

Linux Tutorial



9.1. all files are case sensitive

Files on Linux (or any Unix) are **case sensitive**. This means that **FILE1** is different from **file1**, and **/etc/hosts** is different from **/etc/Hosts** (the latter one does not exist on a typical Linux computer).

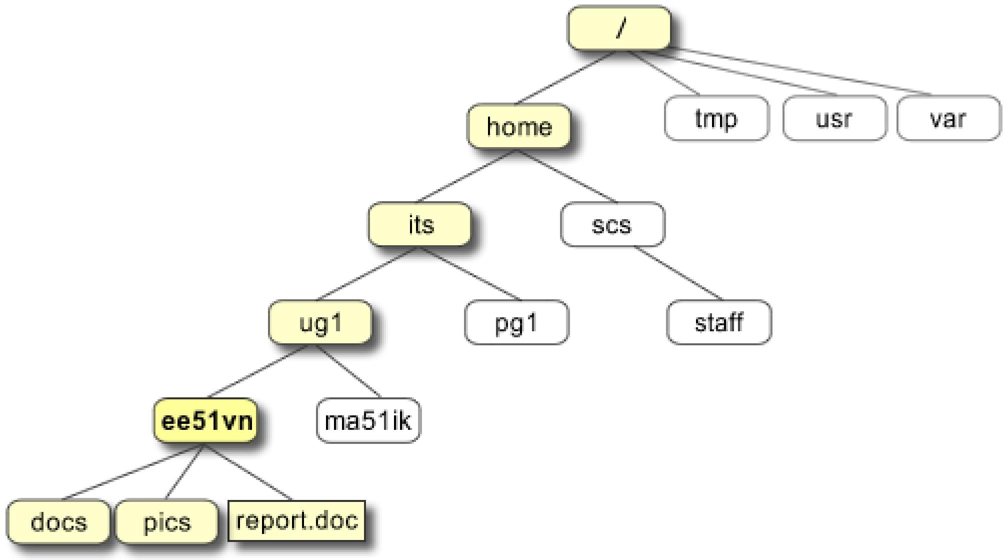
This screenshot shows the difference between two files, one with upper case **W**, the other with lower case **w**.

```
paul@laika:~/Linux$ ls
winter.txt  Winter.txt
paul@laika:~/Linux$ cat winter.txt
It is cold.
paul@laika:~/Linux$ cat Winter.txt
It is very cold!
```

9.2. everything is a file

A **directory** is a special kind of **file**, but it is still a (case sensitive!) **file**. Each terminal window (for example **/dev/pts/4**), any hard disk or partition (for example **/dev/sdb1**) and any process are all represented somewhere in the **file system** as a **file**. It will become clear throughout this course that everything on Linux is a **file**.

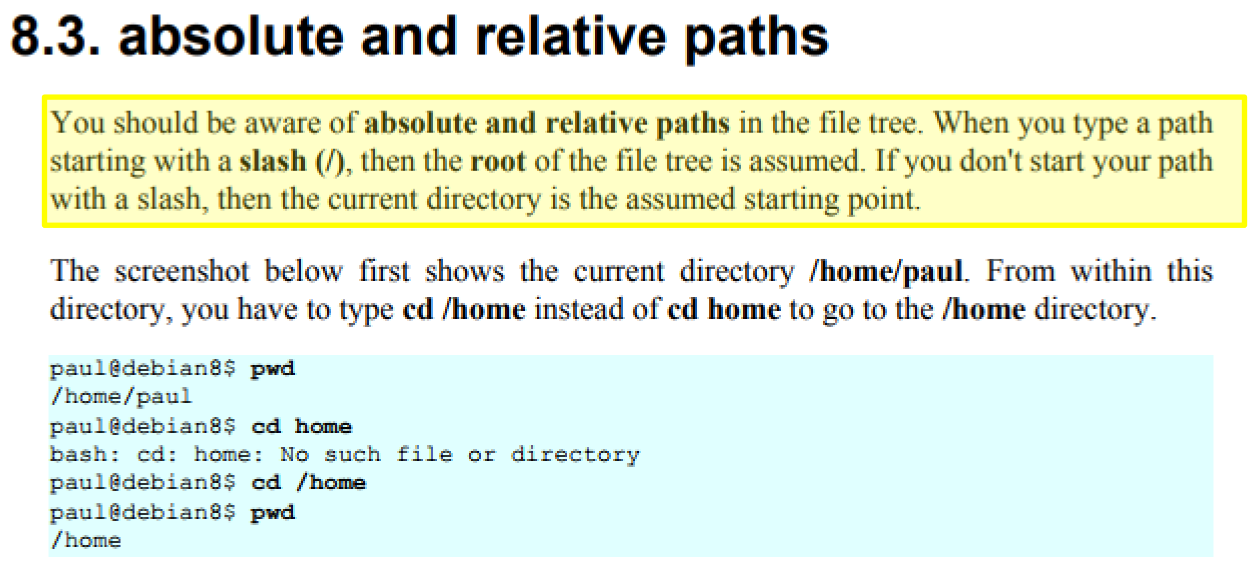
Directory Structure



/home/its/ug1/ee51vn/report.doc

OREGON HEALTH & SCIENCE UNIVERSITY

Absolute/Relative Path



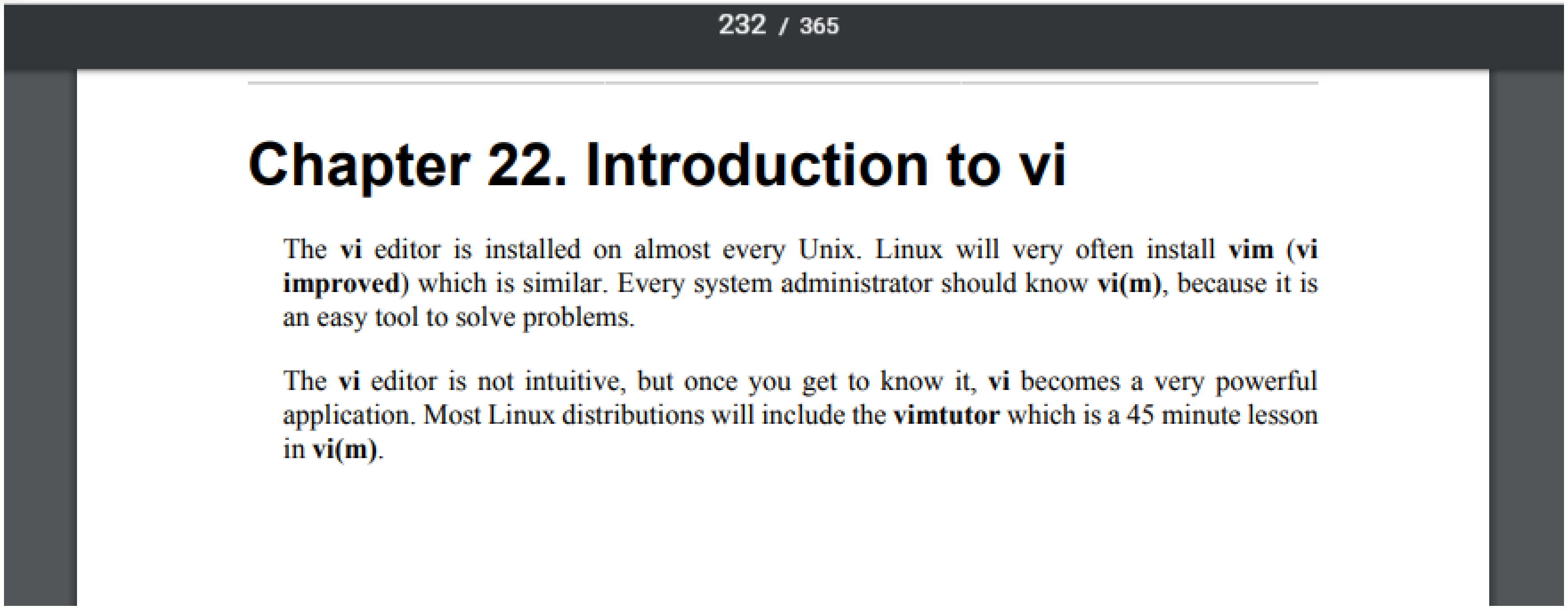
8.3. absolute and relative paths

You should be aware of **absolute and relative paths** in the file tree. When you type a path starting with a **slash (/)**, then the **root** of the file tree is assumed. If you don't start your path with a slash, then the current directory is the assumed starting point.

The screenshot below first shows the current directory **/home/paul**. From within this directory, you have to type **cd /home** instead of **cd home** to go to the **/home** directory.

```
paul@debian8$ pwd
/home/paul
paul@debian8$ cd home
bash: cd: home: No such file or directory
paul@debian8$ cd /home
paul@debian8$ pwd
/home
```

Text Editor: vi

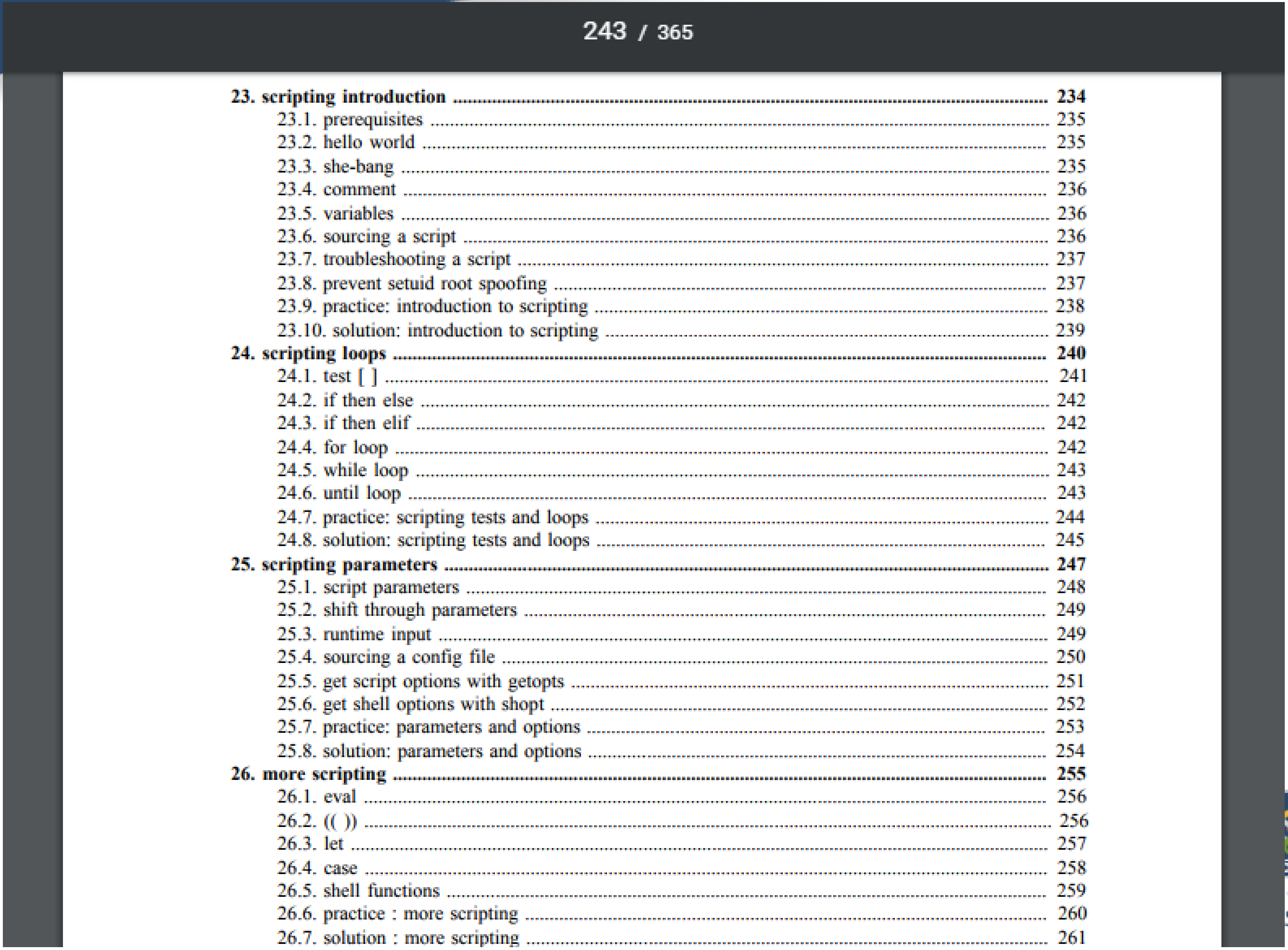

232 / 365

Chapter 22. Introduction to vi

The **vi** editor is installed on almost every Unix. Linux will very often install **vim** (**vi improved**) which is similar. Every system administrator should know **vi(m)**, because it is an easy tool to solve problems.

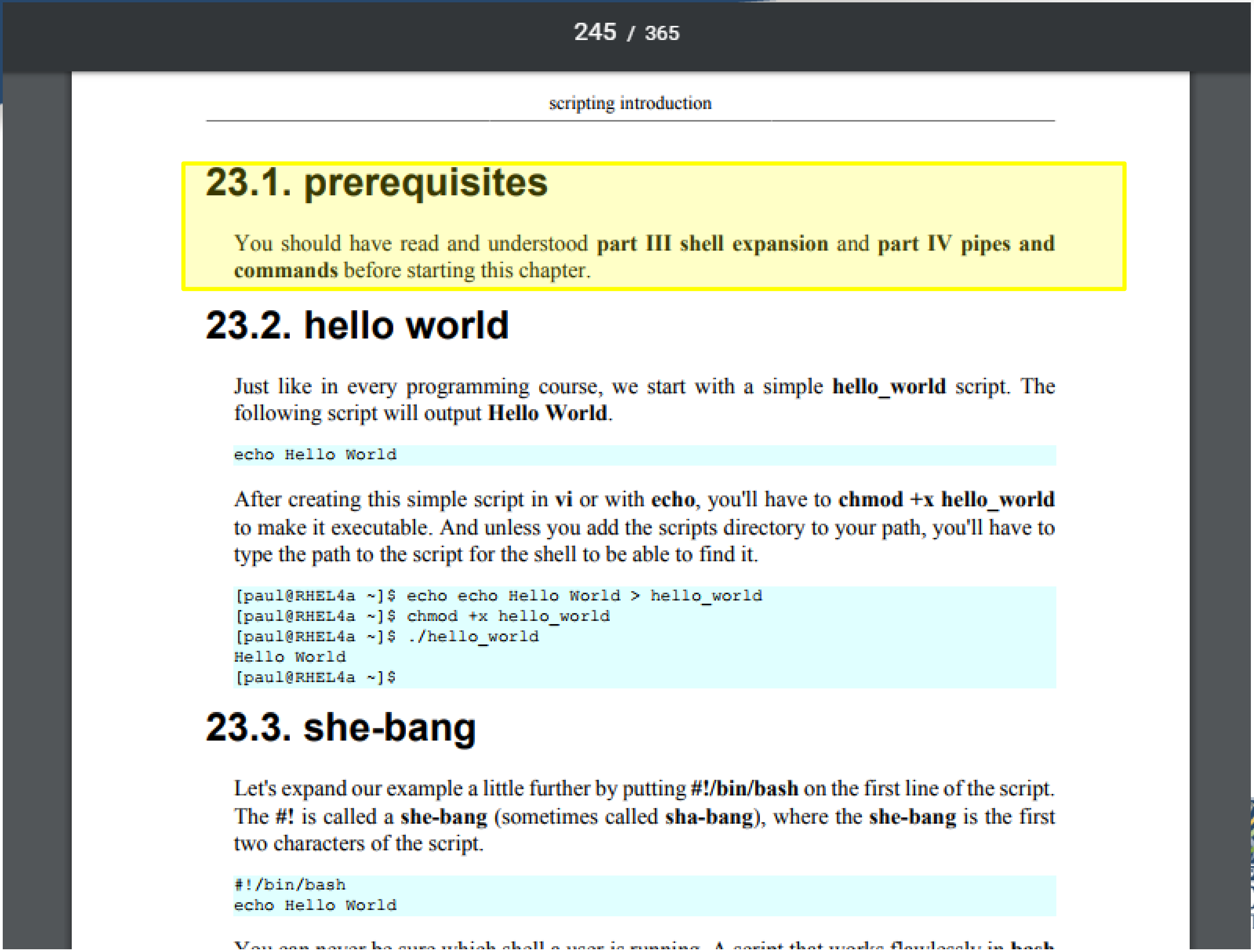
The **vi** editor is not intuitive, but once you get to know it, **vi** becomes a very powerful application. Most Linux distributions will include the **vimtutor** which is a 45 minute lesson in **vi(m)**.

Scripting



243 / 365

Scripting



245 / 365

23.1. prerequisites

You should have read and understood **part III shell expansion** and **part IV pipes and commands** before starting this chapter.

23.2. hello world

Just like in every programming course, we start with a simple **hello_world** script. The following script will output **Hello World**.

```
echo Hello World
```

After creating this simple script in **vi** or with **echo**, you'll have to **chmod +x hello_world** to make it executable. And unless you add the scripts directory to your path, you'll have to type the path to the script for the shell to be able to find it.

```
[paul@RHEL4a ~]$ echo echo Hello World > hello_world
[paul@RHEL4a ~]$ chmod +x hello_world
[paul@RHEL4a ~]$ ./hello_world
Hello World
[paul@RHEL4a ~]$
```

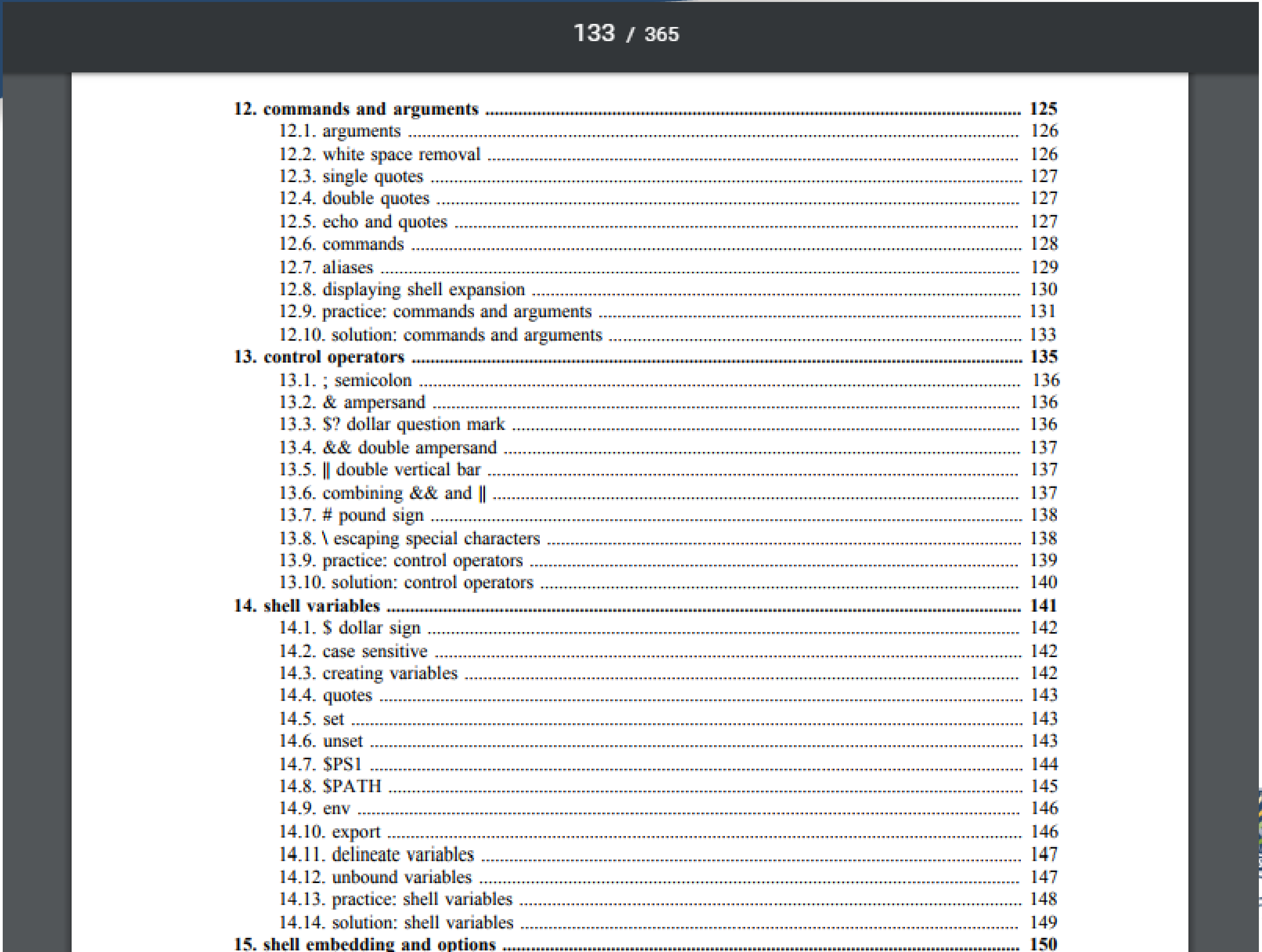
23.3. she-bang

Let's expand our example a little further by putting **#!/bin/bash** on the first line of the script. The **#!** is called a **she-bang** (sometimes called **sha-bang**), where the **she-bang** is the first two characters of the script.

```
#!/bin/bash
echo Hello World
```

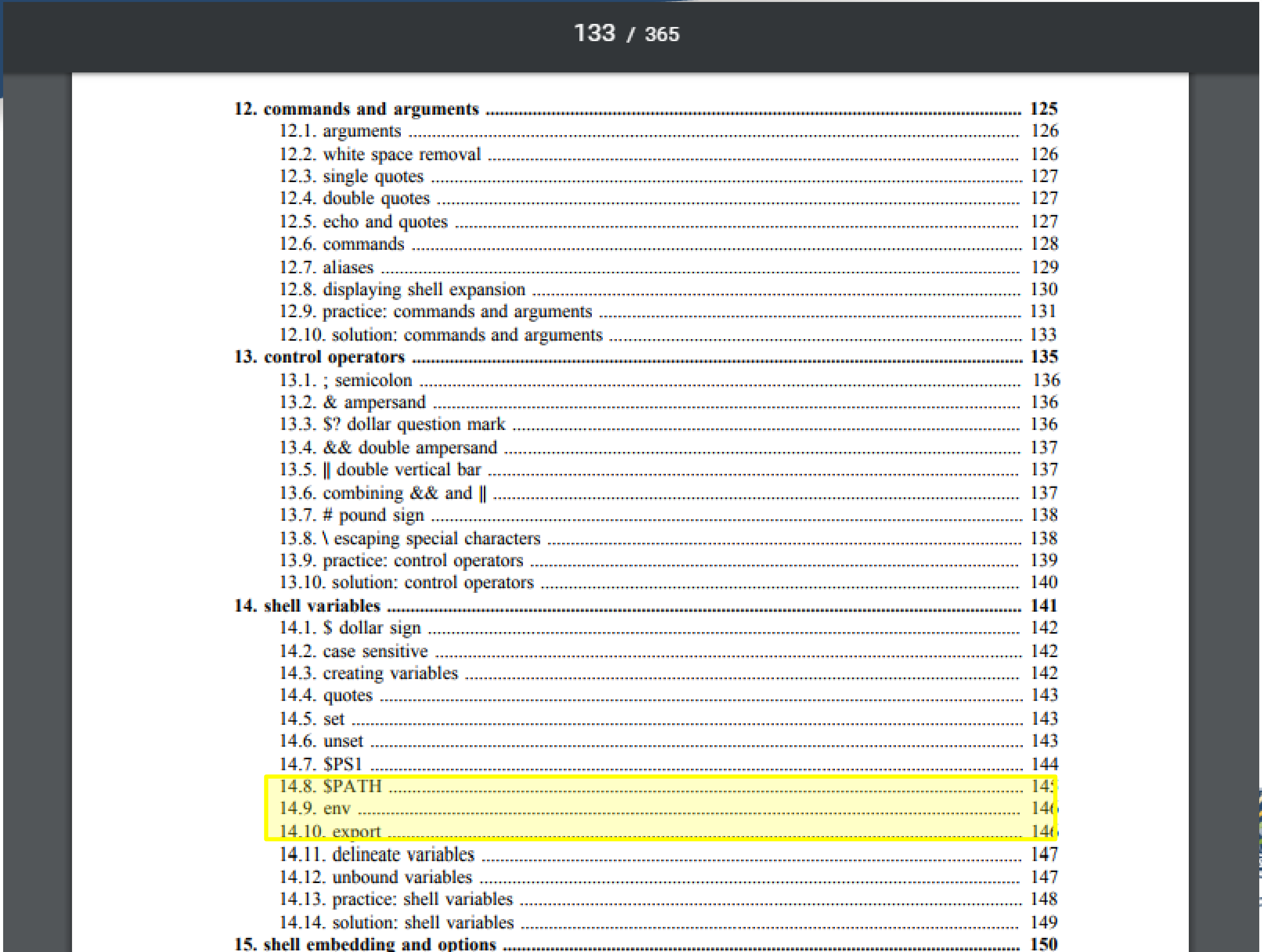
You can never be sure which shell a user is running. A script that works flawlessly in **bash**

Scripting



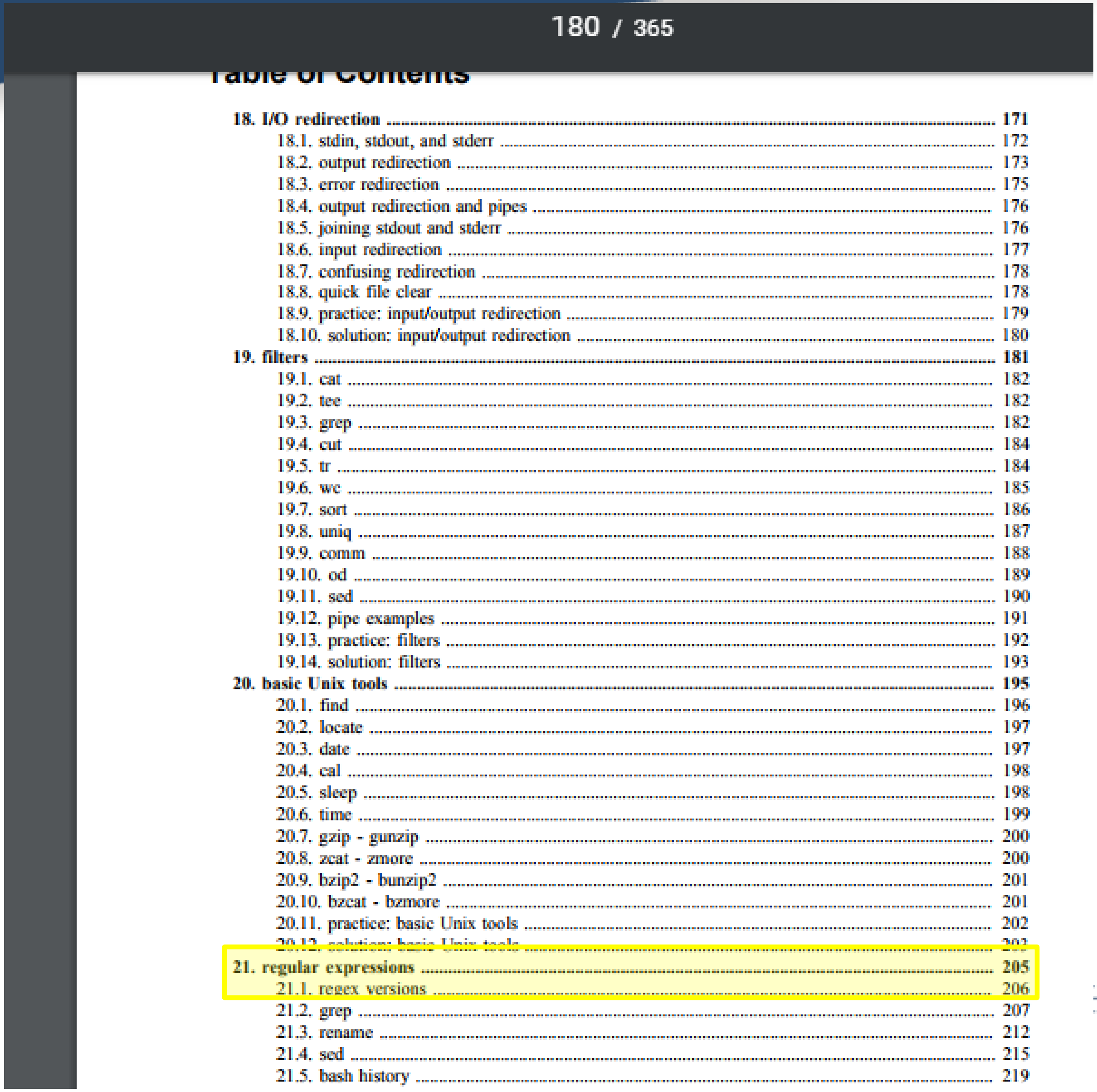
133 / 365

Scripting



133 / 365

Scripting



180 / 365

Table of Contents

Relational Database Design

BMI 535/635

What is a database?

What is a database?

A collection of data organized in a way that can be easily accessed, managed, and updated

Flat Text Files

A database can be as simple as (a collection of) flat files

- organized in a file system
- specific programs to access/update

OREGON HEALTH & SCIENCE UNIVERSITY

Flat Text Files

A database can be as simple as (a collection of) flat files

- organized in a file system
- specific programs to access/update

Disadvantages?

Flat Text Files

A database can be as simple as (a collection of) flat files

- organized in a file system
- specific programs to access/update

Disadvantages?

- lack of access control / security
- lack of data integrity
- lack of flexible/quick access of the data

Database Management Systems (DBMS)

Software that allows for the creation, definition, and manipulation of a database

- accommodate large data sets (storage and querying)
- data consistency and multiple concurrent users
- crash recovery, logging
- security and access control

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements
- data model

Conceptual Database Design

- high level description/constraints of the data (ER Model)

Logical Database Design

- convert conceptual design into a database schema

OREGON HEALTH & SCIENCE UNIVERSITY

Relational Data Model

Relational model

- introduced by E.F Codd in 1970
- collection of one or more relations
- relations represented in tables (rows and columns)
- most commonly used model
    - simple tabular representation
    - permits simple, high level querying of data

OREGON HEALTH & SCIENCE UNIVERSITY

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements
- data model

Conceptual Database Design

- high level description/constraints of the data (ER Model)

Logical Database Design

- convert conceptual design into a database schema

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements
- data model

Database of genes in the human genome

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements
- data model

Database of genes in the human genome

Types of data

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements
- data model

Database of genes in the human genome.

Types of data

- Gene: id, name, chr, type, start, end, strand
- Transcript: id, name, chr, start, end, strand
- Exon: id, chr, start, end, strand

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements
- data model

Database of genes in the human genome.

Types of data

- Gene(id:str, name:str, type:str, chr:str, start:int, end:int, strand:str)
- Transcript(id:str, name:str, chr:str, start:int, end:int, strand:str)
- Exon(id:str, chr:str, start:int, end:int, strand:str)

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements
- data model

Database of genes in the human genome

Types of queries

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements
- data model

Database of genes in the human genome

Types of queries

- How many genes are in the human genome?
- What are the genomic coordinates of “GeneA”?
- What transcripts are associated with of “GeneA”?
- How many exons does “TranscriptA” have?

OREGON HEALTH & SCIENCE UNIVERSITY

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements
- data model

Database of genes in the human genome

Performance requirements

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements
- data model

Database of genes in the human genome

Performance requirements

- large number of genes
- large number of transcripts
- types of queries
- ensure (frequent) queries are efficient

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements
- data model

Conceptual Database Design

- high level description/constraints of the data (ER Model)

Logical Database Design

- convert conceptual design into a database schema

OREGON HEALTH & SCIENCE UNIVERSITY

Entity-Relationship (ER) Model

Entity

- describes real world objects
- i.e., students, hospitals, genes, transcripts, exons

Relationship

- association between two or more entities
- i.e., in, works_in, part_of, associated_with

Database Design Considerations

Requirements analysis

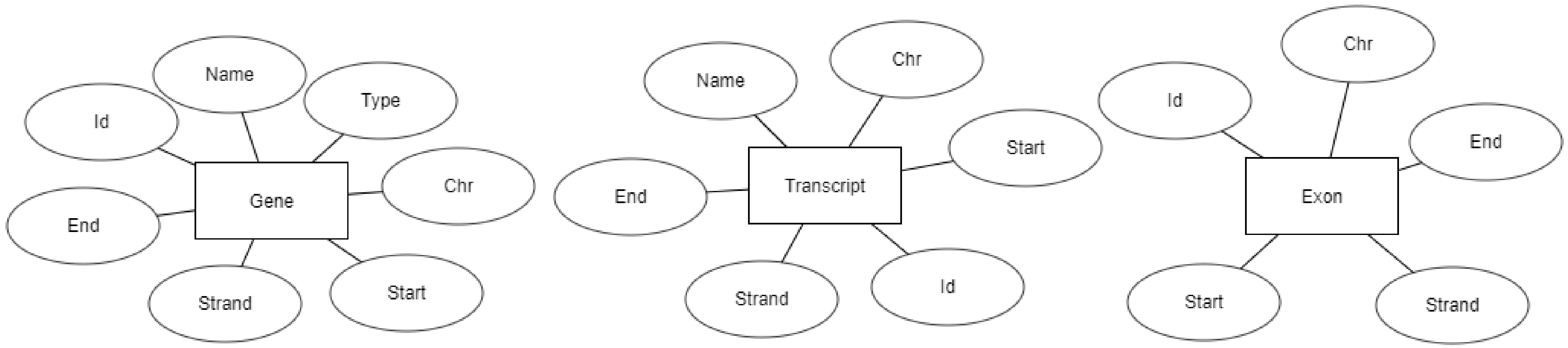
- type of data, types of queries, performance requirements
- data model

Database of genes in the human genome.

Types of data

- Gene(Id:str, Name:str, Type:str, Chr:str, Start:int, End:int, Strand:str)
- Transcript(Id:str, Name:str, Chr:str, Start:int, End:int, Strand:str)
- Exon(Id:str, Chr:str, Start:int, End:int, Strand:str)

ER Model: Entity



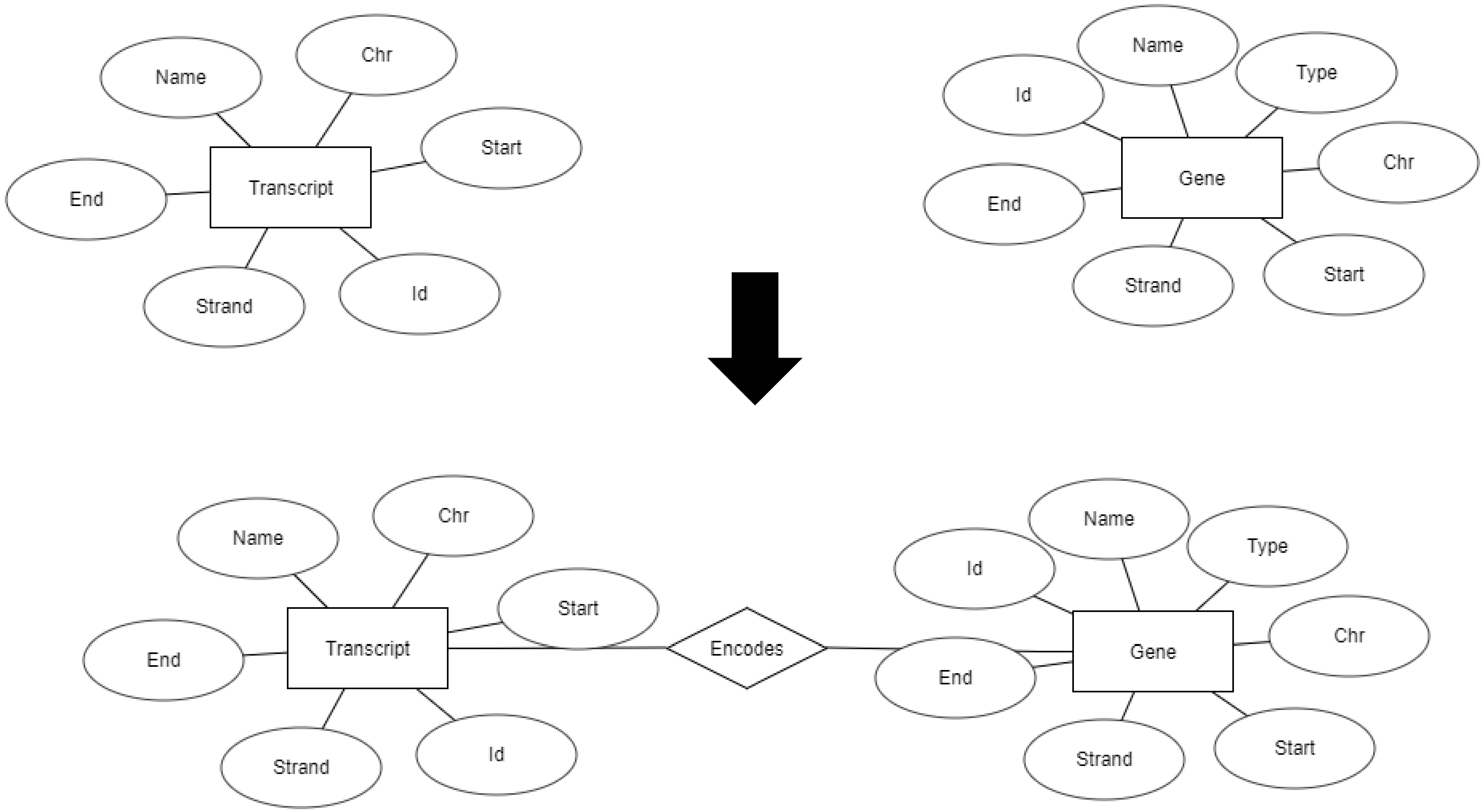
Database of genes in the human genome.

Types of data

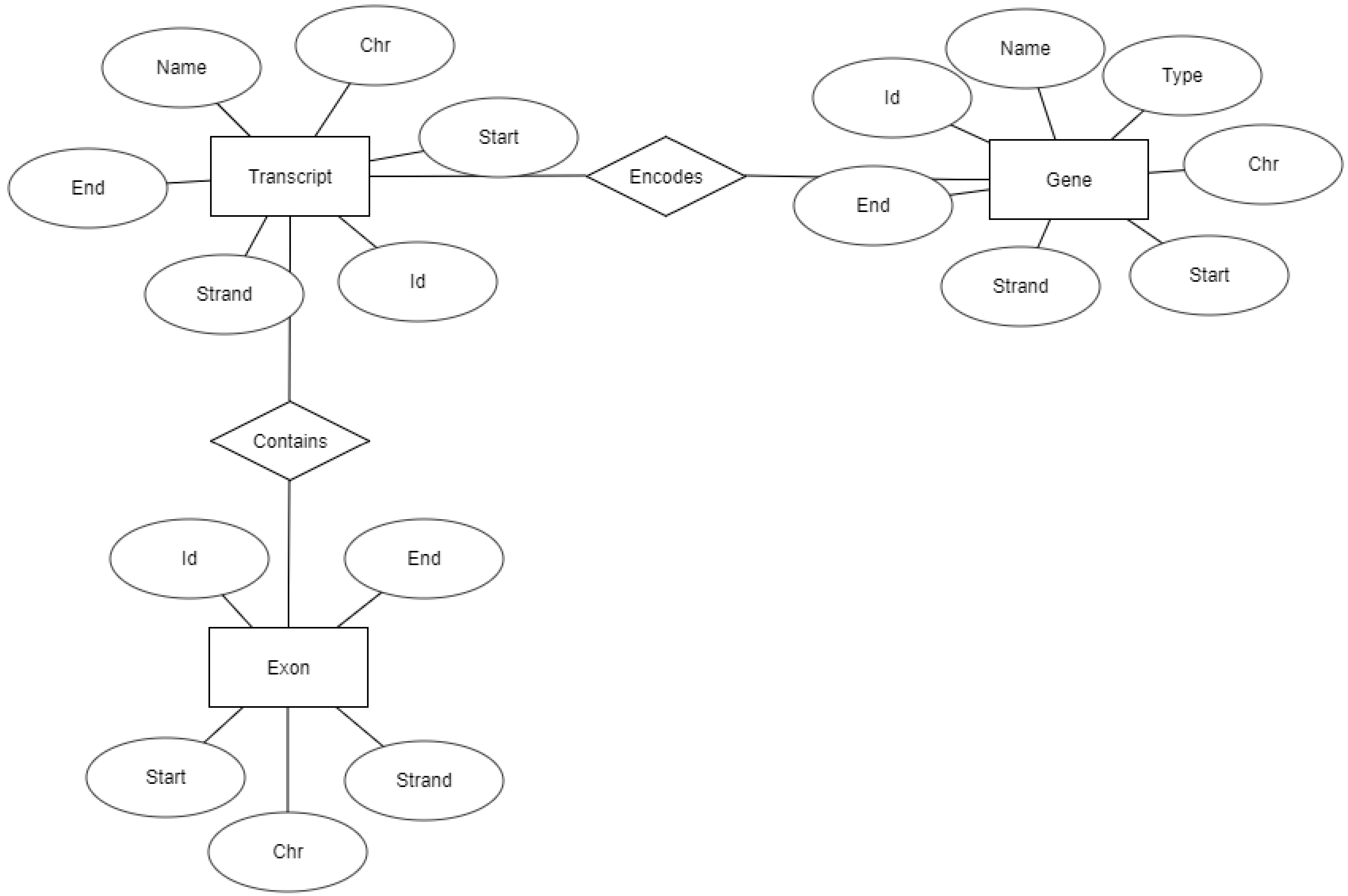
- Gene(Id:str, Name:str, Type:str, Chr:str, Start:int, End:int, Strand:str)
- Transcript(Id:str, Name:str, Chr:str, Start:int, End:int, Strand:str)
- Exon(Id:str, Chr:str, Start:int, End:int, Strand:str)

OREGON HEALTH & SCIENCE UNIVERSITY

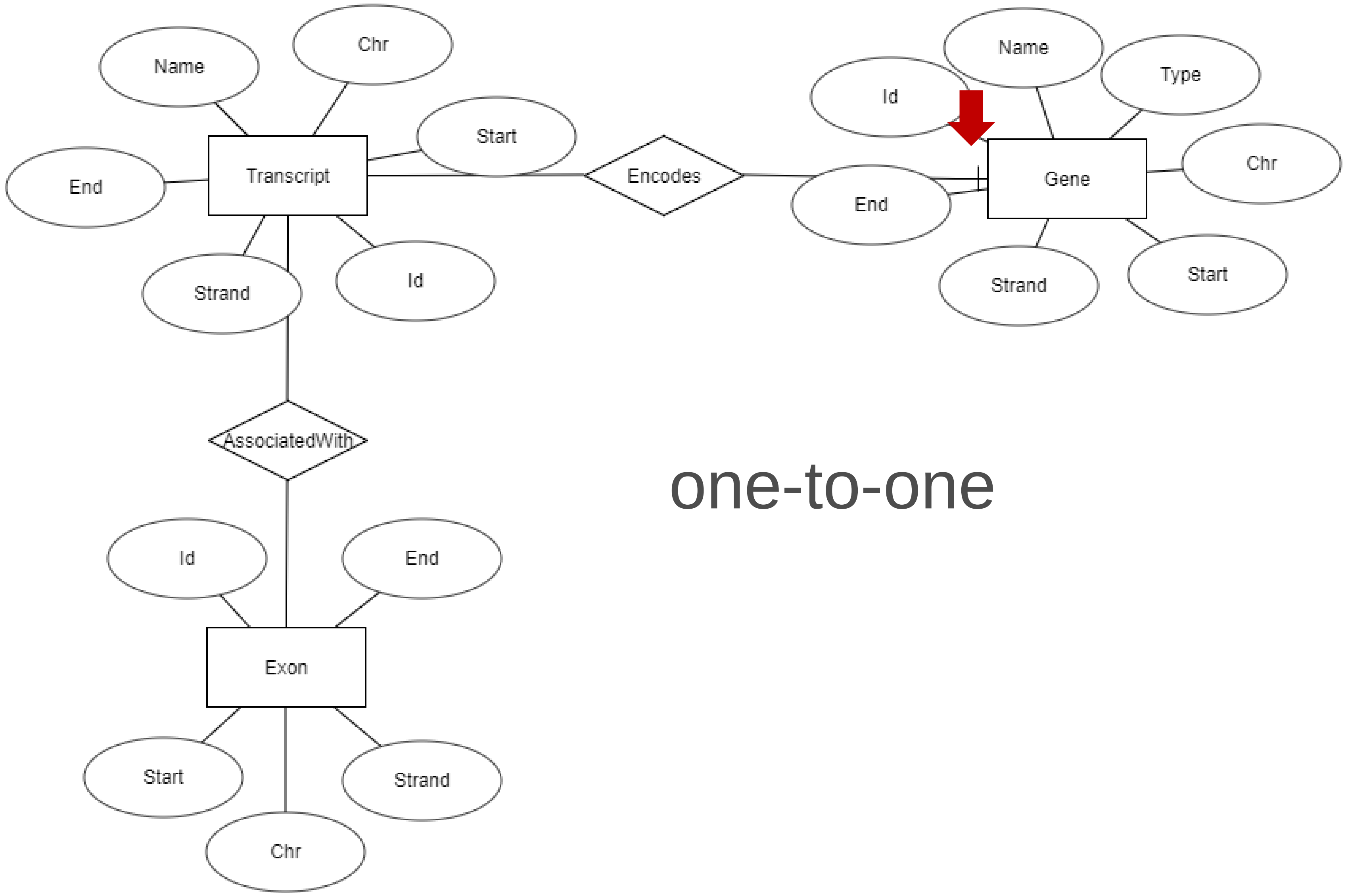
ER Model: Relationship



ER Model: Relationship

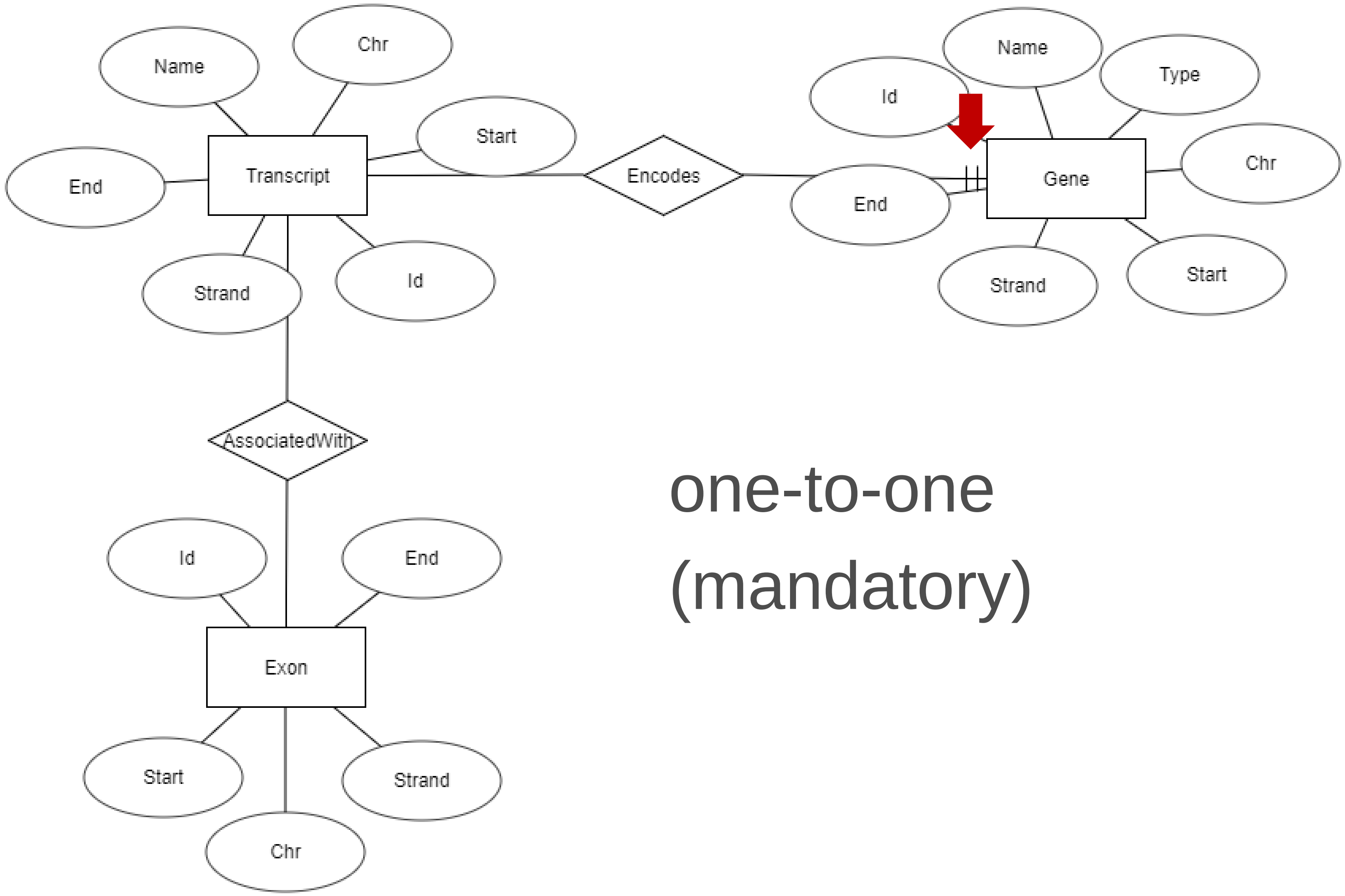


ER Model: Cardinality



one-to-one

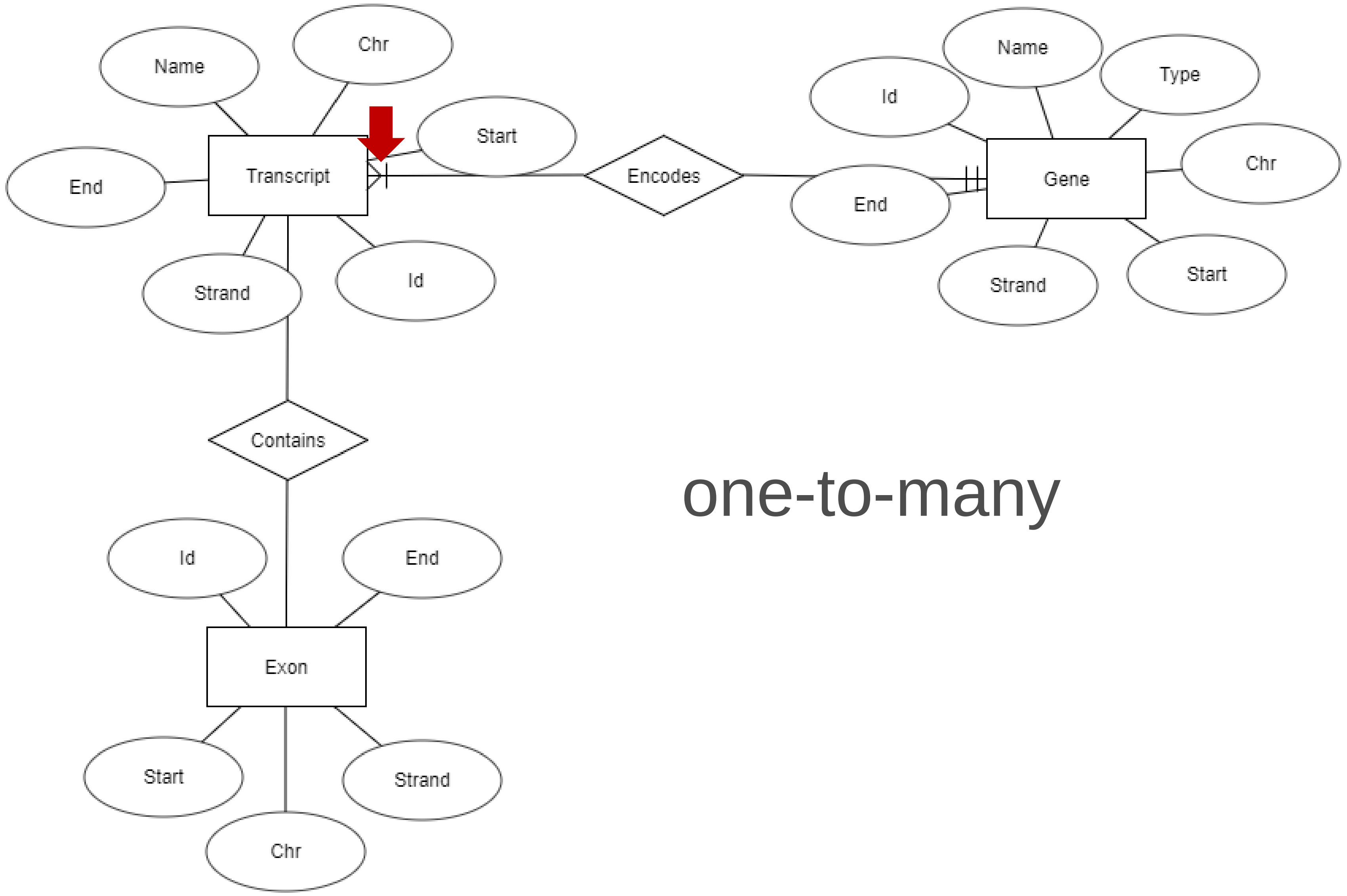
ER Model: Cardinality



one-to-one

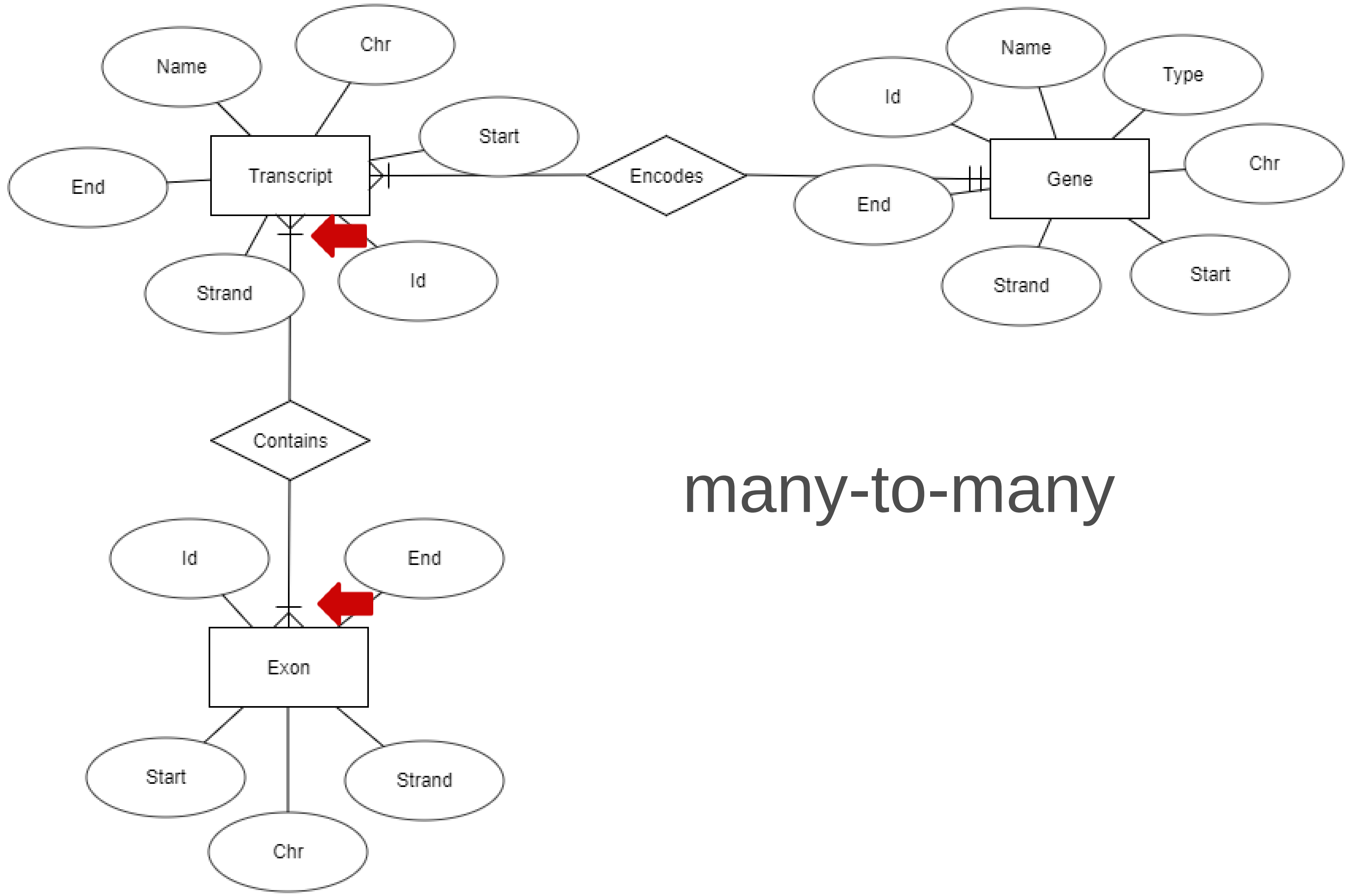
(mandatory)

ER Model: Cardinality



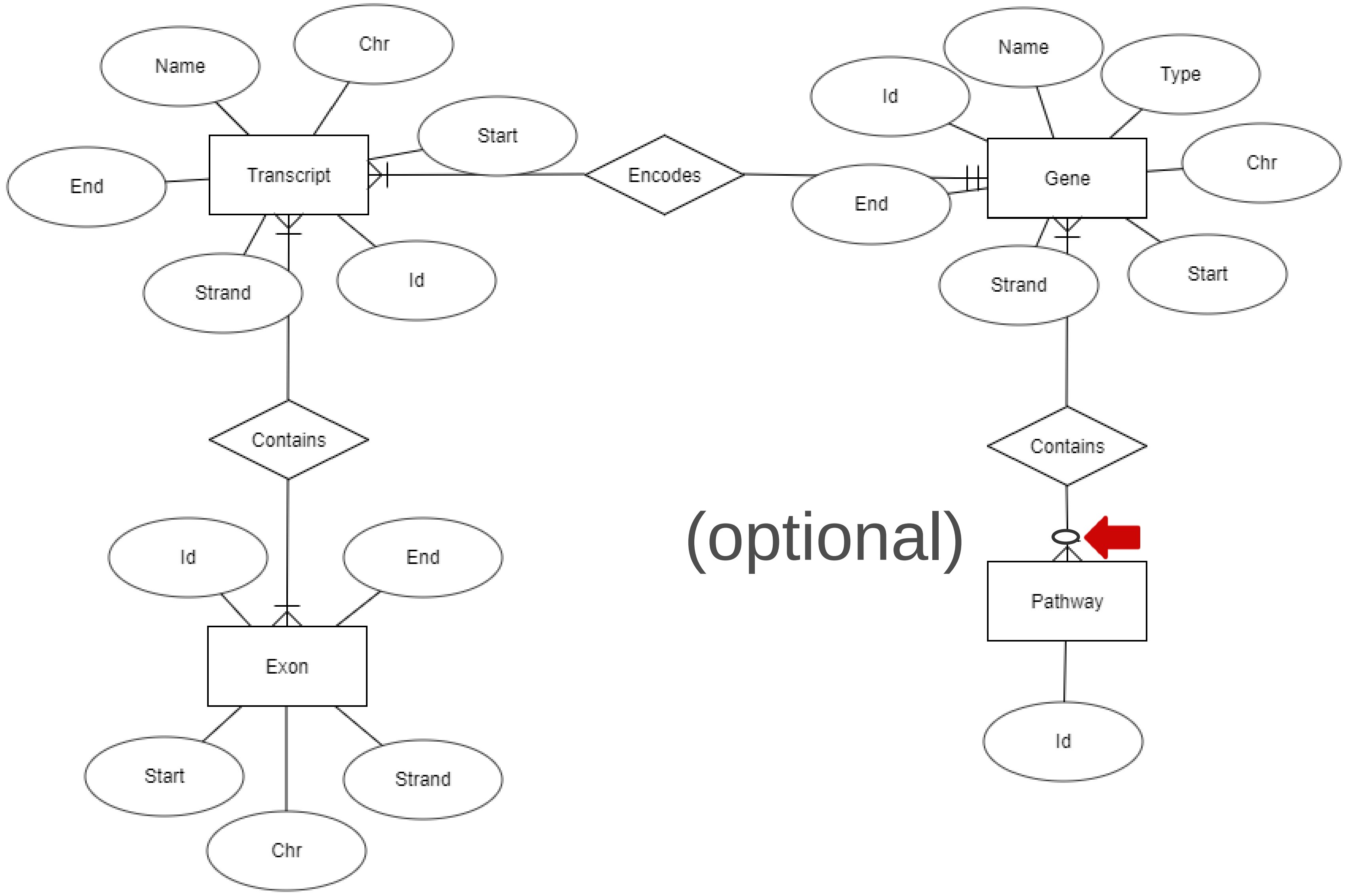
one-to-many

ER Model: Cardinality



many-to-many

ER Model



(optional)

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements

Conceptual Database Design

- high level description/constraints of the data (ER Model)

Logical Database Design

- convert conceptual design into a database schema

RDMS

PostgreSQL

MySQL™

Microsoft® SQL Server®

ORACLE®

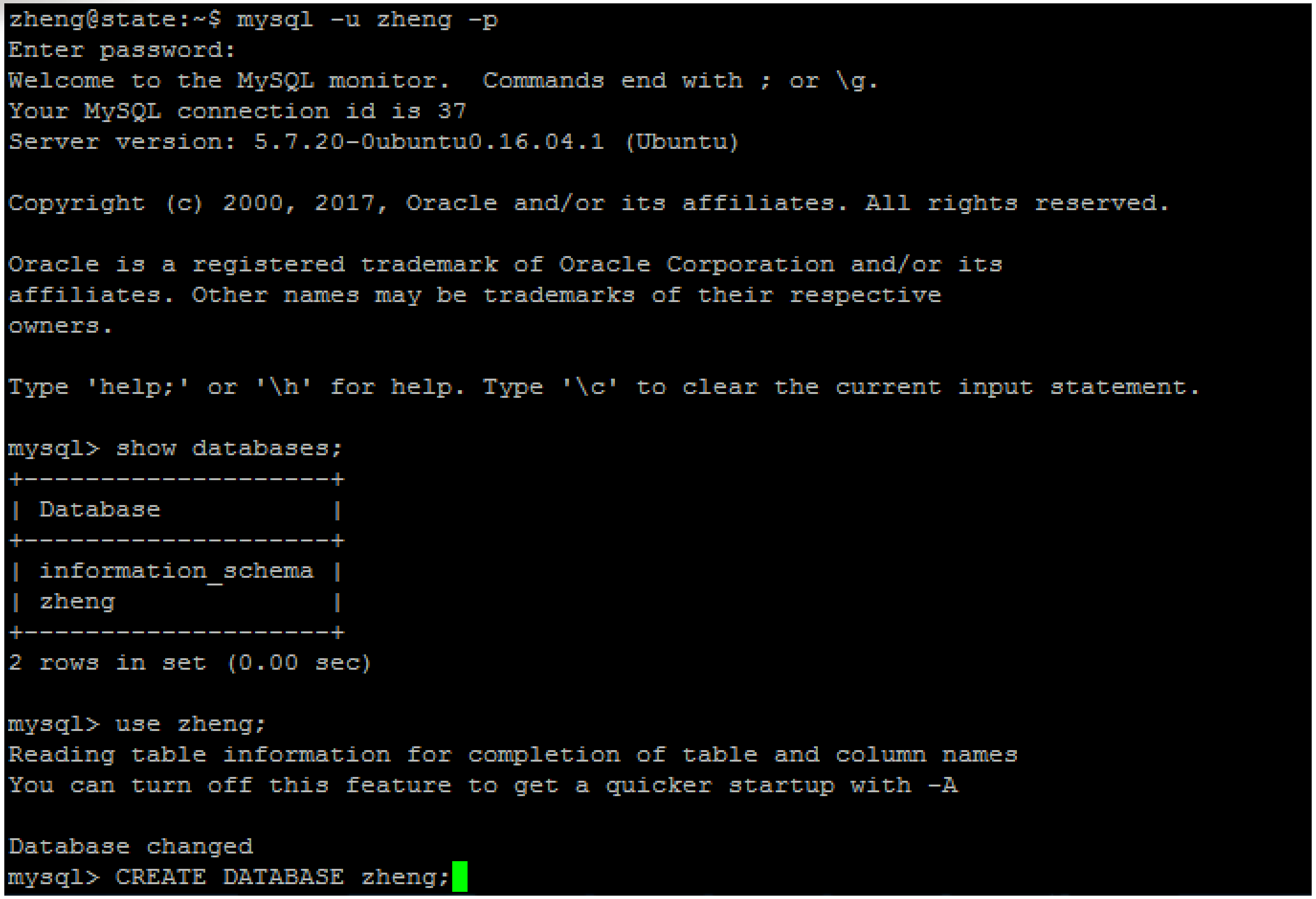
Structured Query Language (SQL)

Widely used language for creating , manipulating, and querying relational databases.

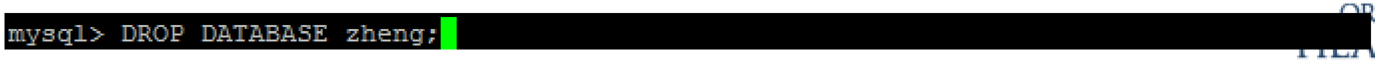
American National Standards Institute (ANSI)

- SQL-86
- SQL-92
- SQL-99

MySQL on State



```
zheng@state:~$ mysql -u zheng -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37
Server version: 5.7.20-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| zheng              |
+--------------------+
2 rows in set (0.00 sec)

mysql> use zheng;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> CREATE DATABASE zheng;
```

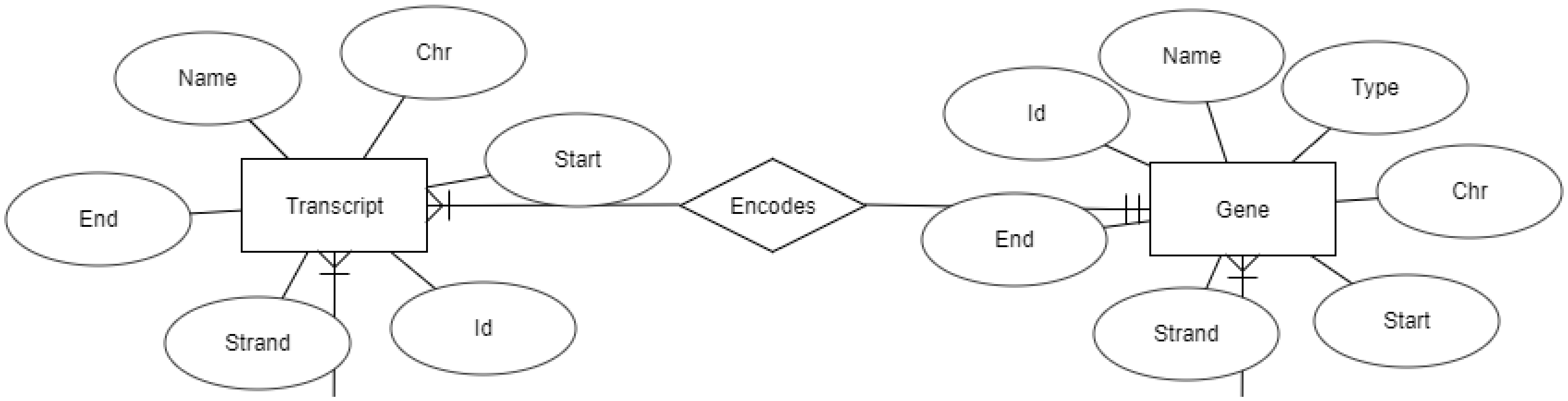


```
mysql> DROP DATABASE zheng;
```

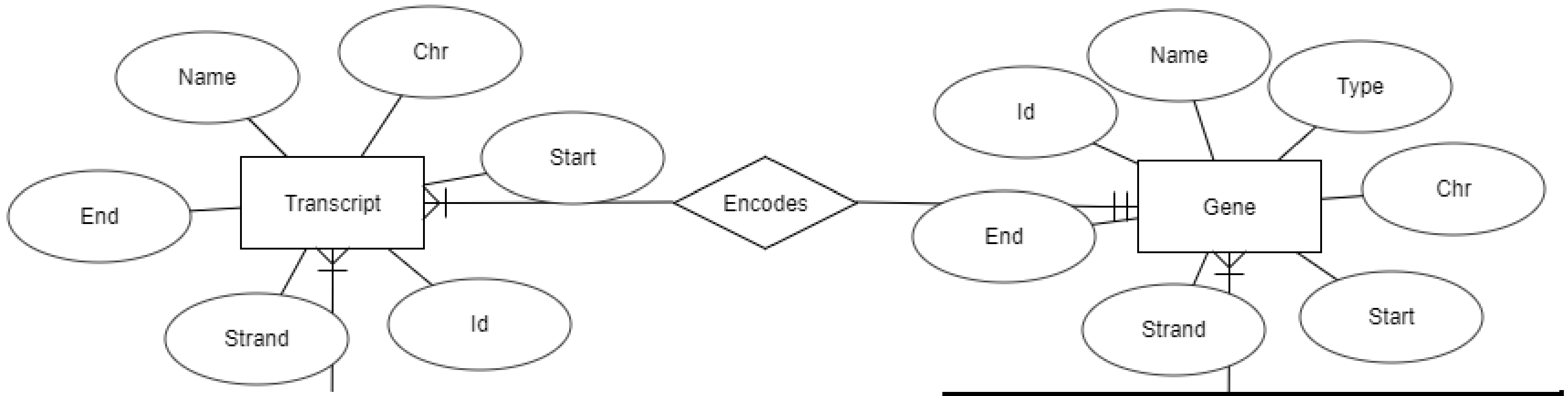
RDMS: Tables

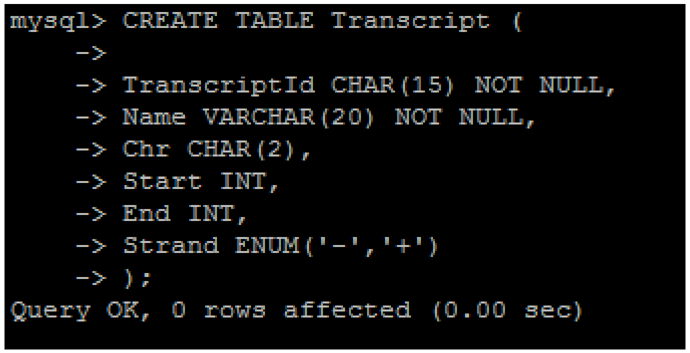
Table Schema

- describes table name, name of each column (field), data domain of column

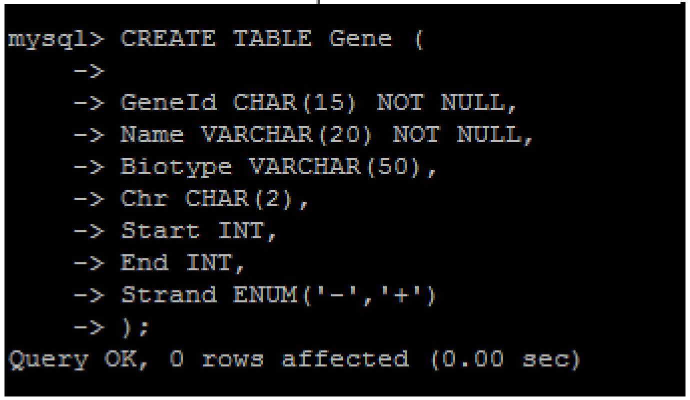


SQL: CREATE TABLE



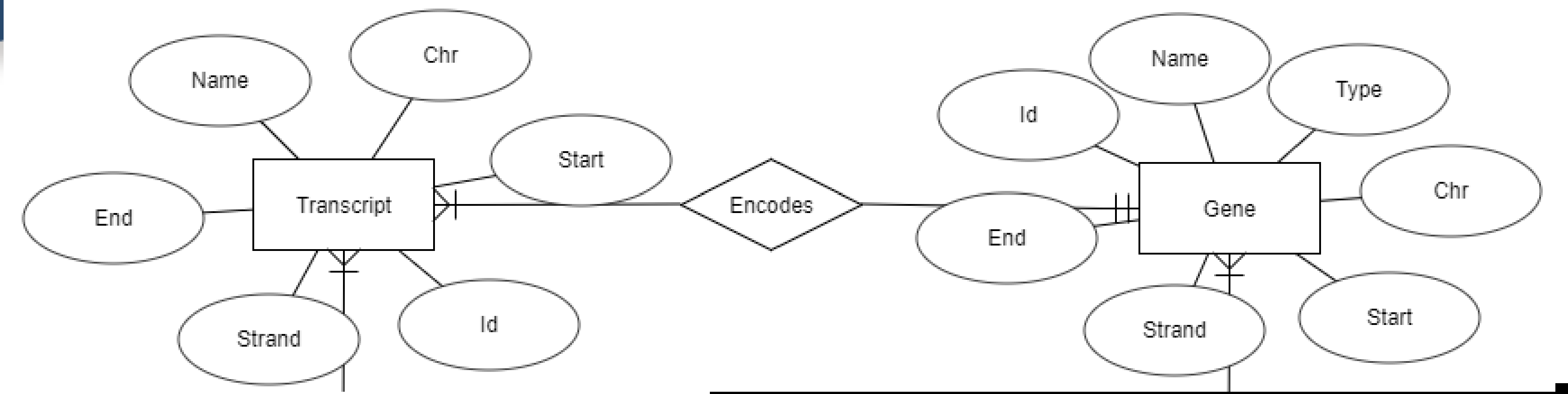


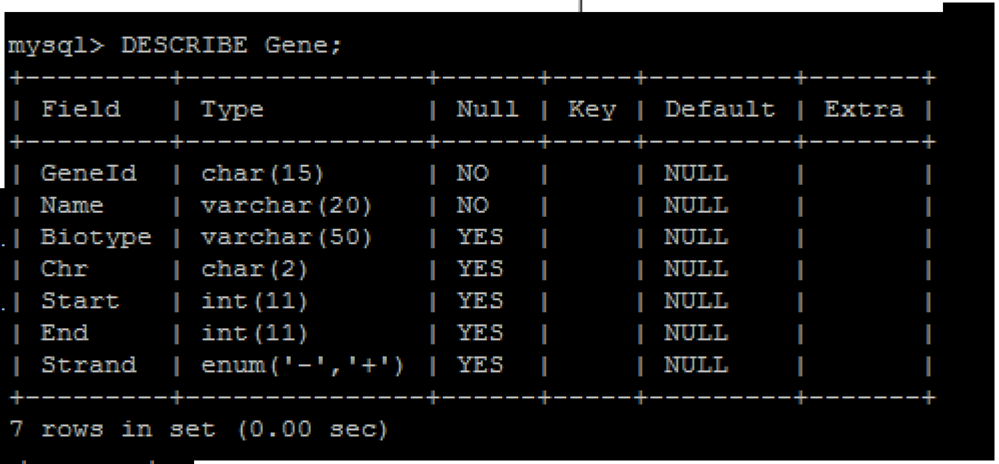
```
mysql> CREATE TABLE Transcript (
    ->
    -> TranscriptId CHAR(15) NOT NULL,
    -> Name VARCHAR(20) NOT NULL,
    -> Chr CHAR(2),
    -> Start INT,
    -> End INT,
    -> Strand ENUM('-','+')
    -> );
Query OK, 0 rows affected (0.00 sec)
```



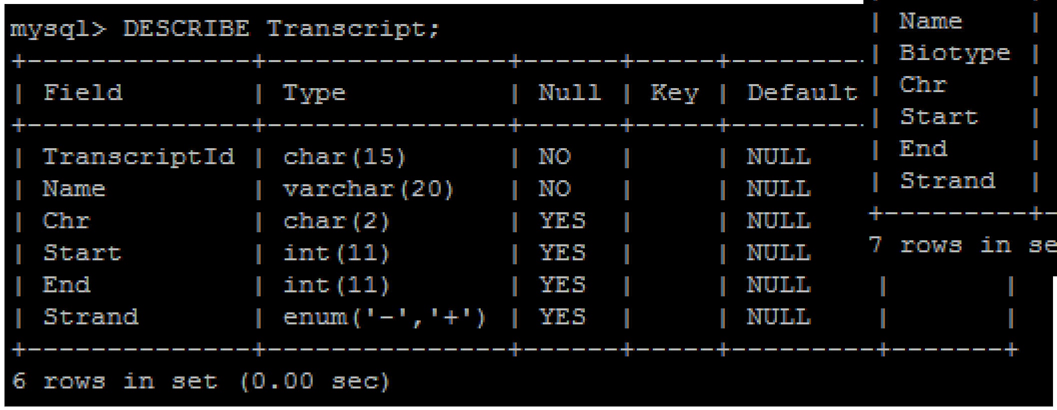
```
mysql> CREATE TABLE Gene (
    ->
    -> GeneId CHAR(15) NOT NULL,
    -> Name VARCHAR(20) NOT NULL,
    -> Biotype VARCHAR(50),
    -> Chr CHAR(2),
    -> Start INT,
    -> End INT,
    -> Strand ENUM('-','+')
    -> );
Query OK, 0 rows affected (0.00 sec)
```

SQL: DESCRIBE TABLE



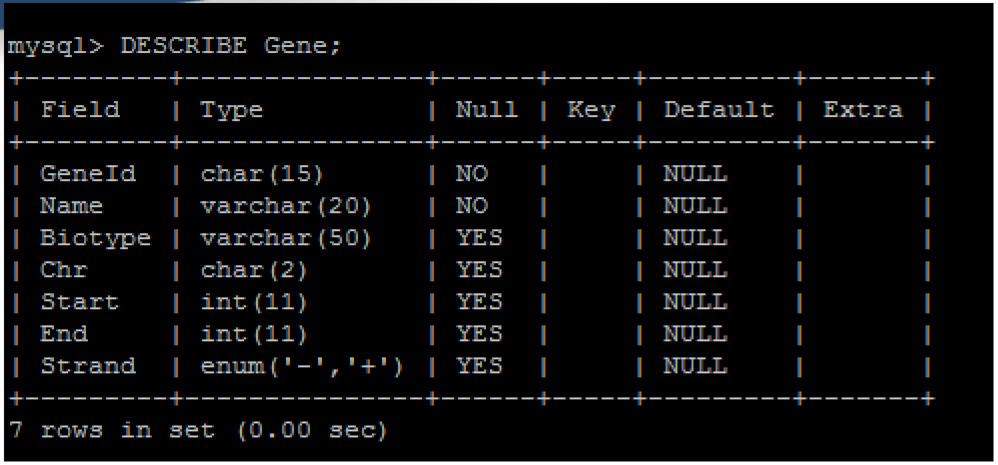


```
mysql> DESCRIBE Gene;
+----------+----------------+------+-----+---------+-------+
| Field    | Type           | Null | Key | Default | Extra |
+----------+----------------+------+-----+---------+-------+
| GeneId   | char(15)       | NO   |     | NULL    |       |
| Name     | varchar(20)    | NO   |     | 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)
```

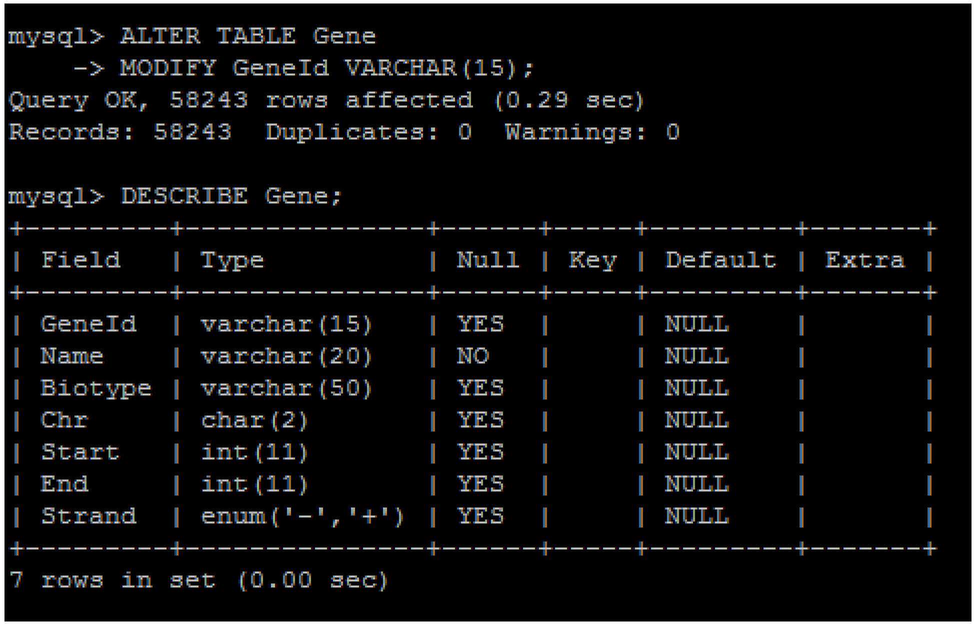


```
mysql> DESCRIBE Transcript;
+--------------+---------------+------+-----+--------
| Field        | Type          | Null | Key | Default
+--------------+---------------+------+-----+--------
| TranscriptId | char(15)      | NO   |     | NULL
| Name         | varchar(20)   | NO   |     | NULL
| Chr          | char(2)       | YES  |     | NULL
| Start        | int(11)       | YES  |     | NULL
| End          | int(11)       | YES  |     | NULL    |       |
| Strand       | enum('-','+') | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

SQL: ALTER TABLE

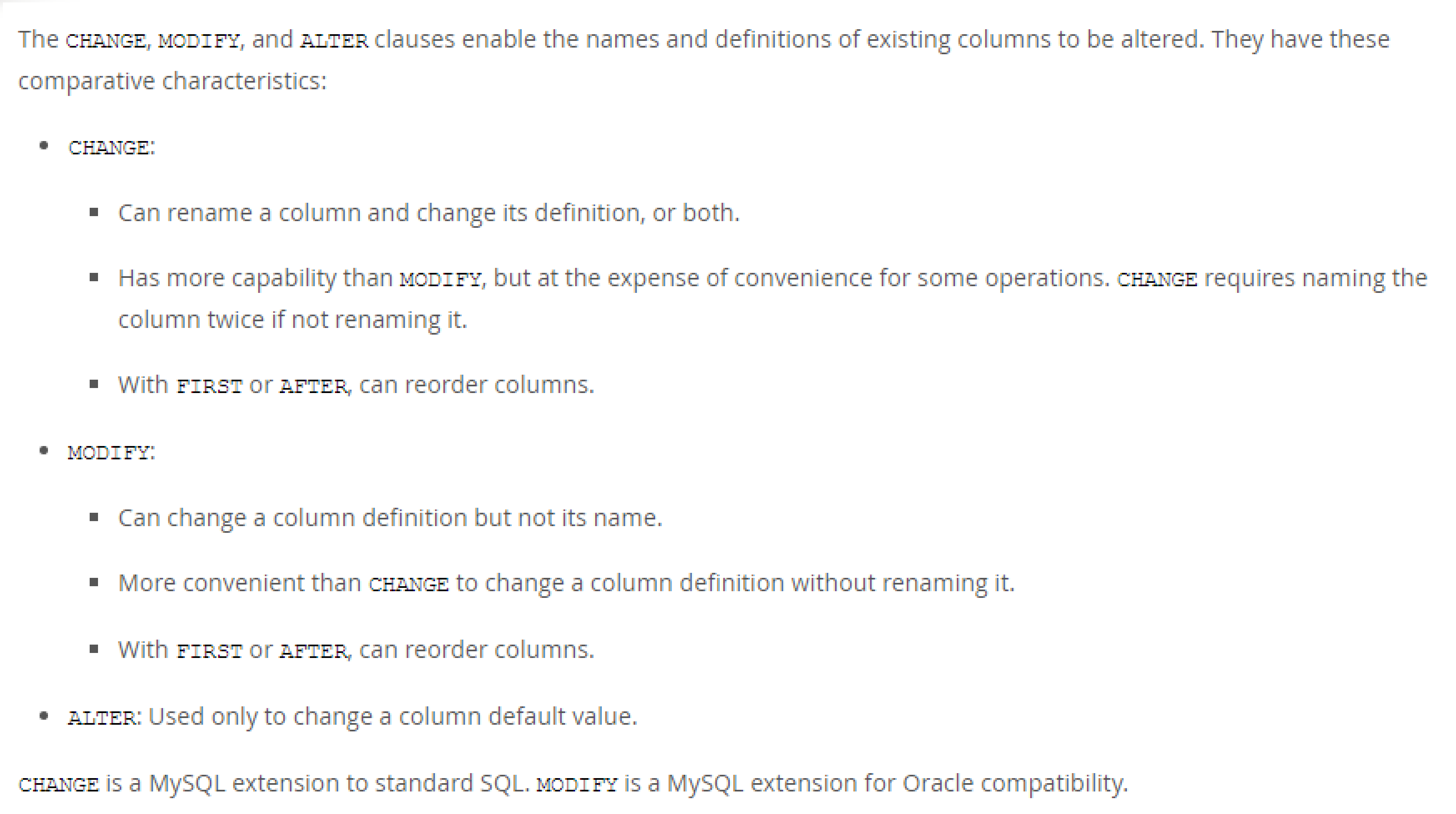


```
mysql> DESCRIBE Gene;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| GeneId   | char(15)      | NO   |     | NULL    |       |
| Name     | varchar(20)   | NO   |     | 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> ALTER TABLE Gene
    -> MODIFY GeneId VARCHAR(15);
Query OK, 58243 rows affected (0.29 sec)
Records: 58243  Duplicates: 0  Warnings: 0

mysql> DESCRIBE Gene;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| GeneId   | varchar(15)   | YES  |     | 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)
```

SQL: ALTER TABLE



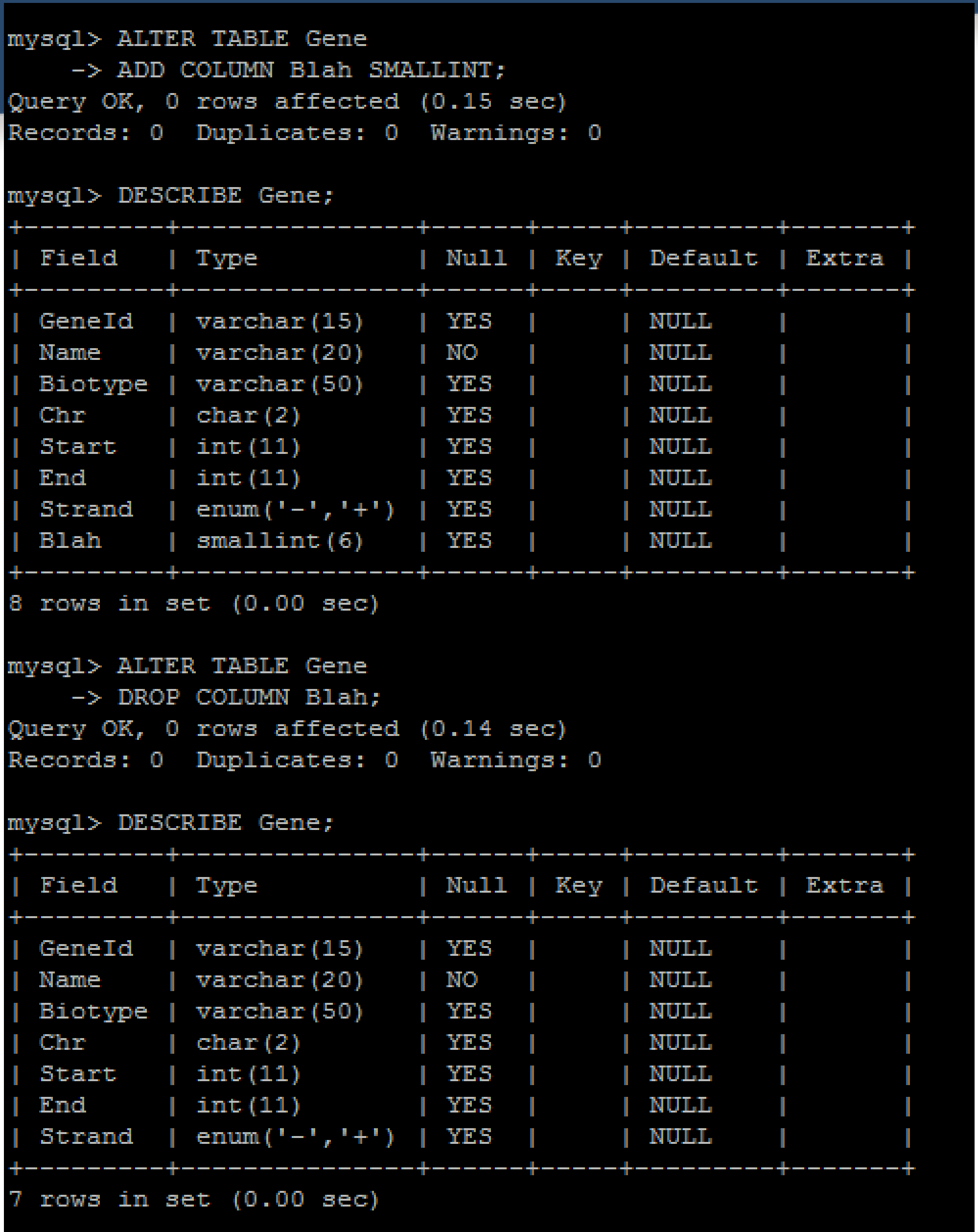
The `CHANGE`, `MODIFY`, and `ALTER` clauses enable the names and definitions of existing columns to be altered. They have these comparative characteristics:

- `CHANGE`:
  - Can rename a column and change its definition, or both.
  - Has more capability than `MODIFY`, but at the expense of convenience for some operations. `CHANGE` requires naming the column twice if not renaming it.
  - With `FIRST` or `AFTER`, can reorder columns.
- `MODIFY`:
  - Can change a column definition but not its name.
  - More convenient than `CHANGE` to change a column definition without renaming it.
  - With `FIRST` or `AFTER`, can reorder columns.
- `ALTER`: Used only to change a column default value.

`CHANGE` is a MySQL extension to standard SQL. `MODIFY` is a MySQL extension for Oracle compatibility.

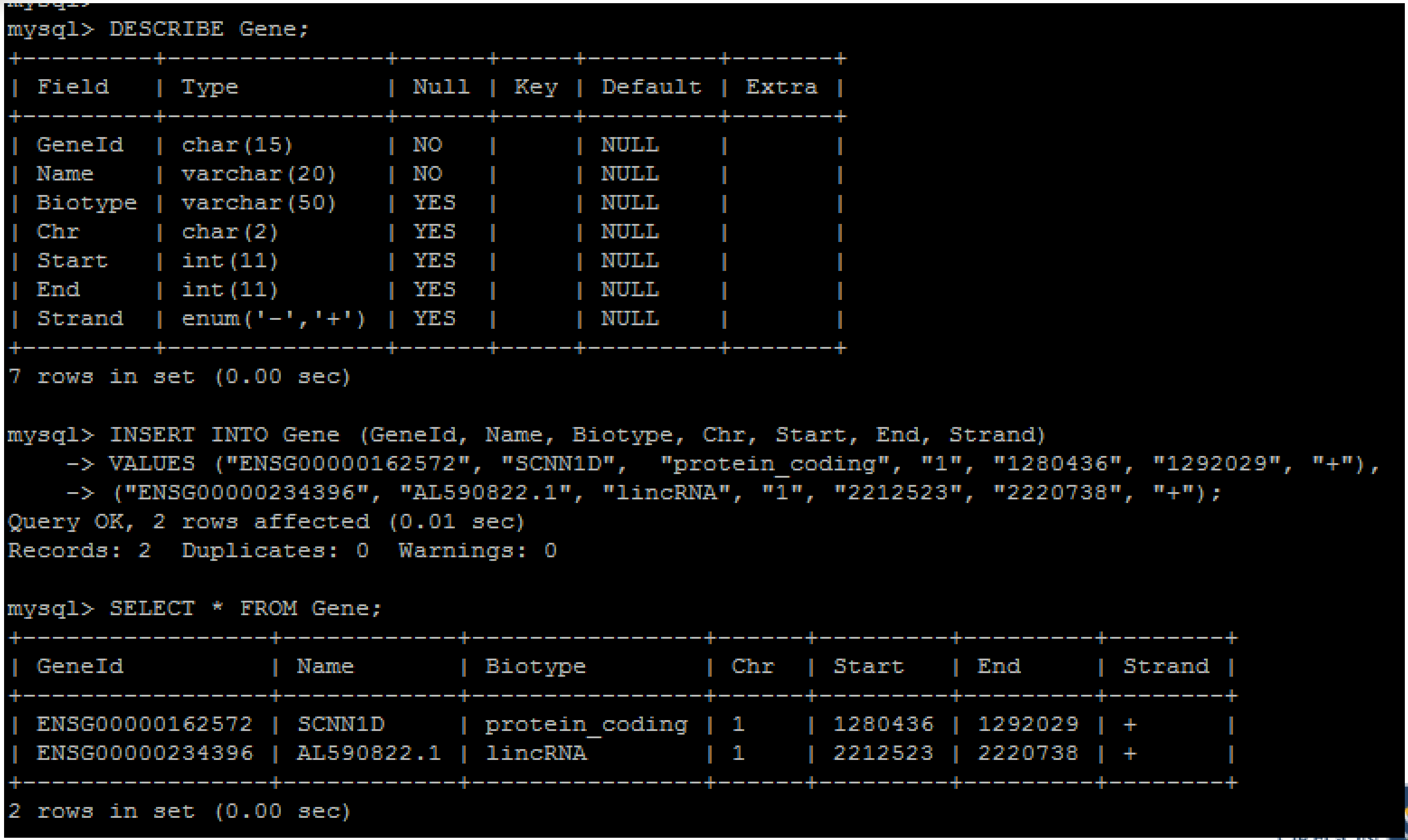
https://dev.mysql.com/doc/refman/5.7/en/alter-table.html

SQL: ALTER TABLE



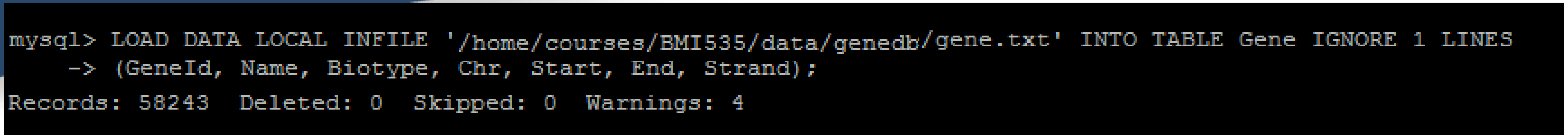
```
mysql> ALTER TABLE Gene
    -> ADD COLUMN Blah SMALLINT;
Query OK, 0 rows affected (0.15 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE Gene;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| GeneId   | varchar(15)   | YES  |     | 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    |       |
| Blah     | smallint(6)   | YES  |     | NULL    |       |
+----------+---------------+------+-----+---------+-------+
8 rows in set (0.00 sec)

mysql> ALTER TABLE Gene
    -> DROP COLUMN Blah;
Query OK, 0 rows affected (0.14 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE Gene;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| GeneId   | varchar(15)   | YES  |     | 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)
```

SQL: INSERT INTO



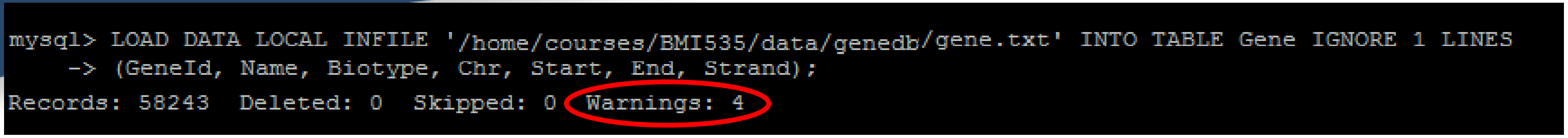
```
mysql> DESCRIBE Gene;
+----------+----------------+------+-----+---------+-------+
| Field    | Type           | Null | Key | Default | Extra |
+----------+----------------+------+-----+---------+-------+
| GeneId   | char(15)       | NO   |     | NULL    |       |
| Name     | varchar(20)    | NO   |     | 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> INSERT INTO Gene (GeneId, Name, Biotype, Chr, Start, End, Strand)
    -> VALUES ("ENSG00000162572", "SCNN1D",  "protein_coding", "1", "1280436", "1292029", "+"),
    -> ("ENSG00000234396", "AL590822.1", "lincRNA", "1", "2212523", "2220738", "+");
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Gene;
+-----------------+------------+----------------+------+---------+---------+--------+
| GeneId          | Name       | Biotype        | Chr  | Start   | End     | Strand |
+-----------------+------------+----------------+------+---------+---------+--------+
| ENSG00000162572 | SCNN1D     | protein_coding | 1    | 1280436 | 1292029 | +      |
| ENSG00000234396 | AL590822.1 | lincRNA        | 1    | 2212523 | 2220738 | +      |
+-----------------+------------+----------------+------+---------+---------+--------+
2 rows in set (0.00 sec)
```

SQL: Loading Data From a File



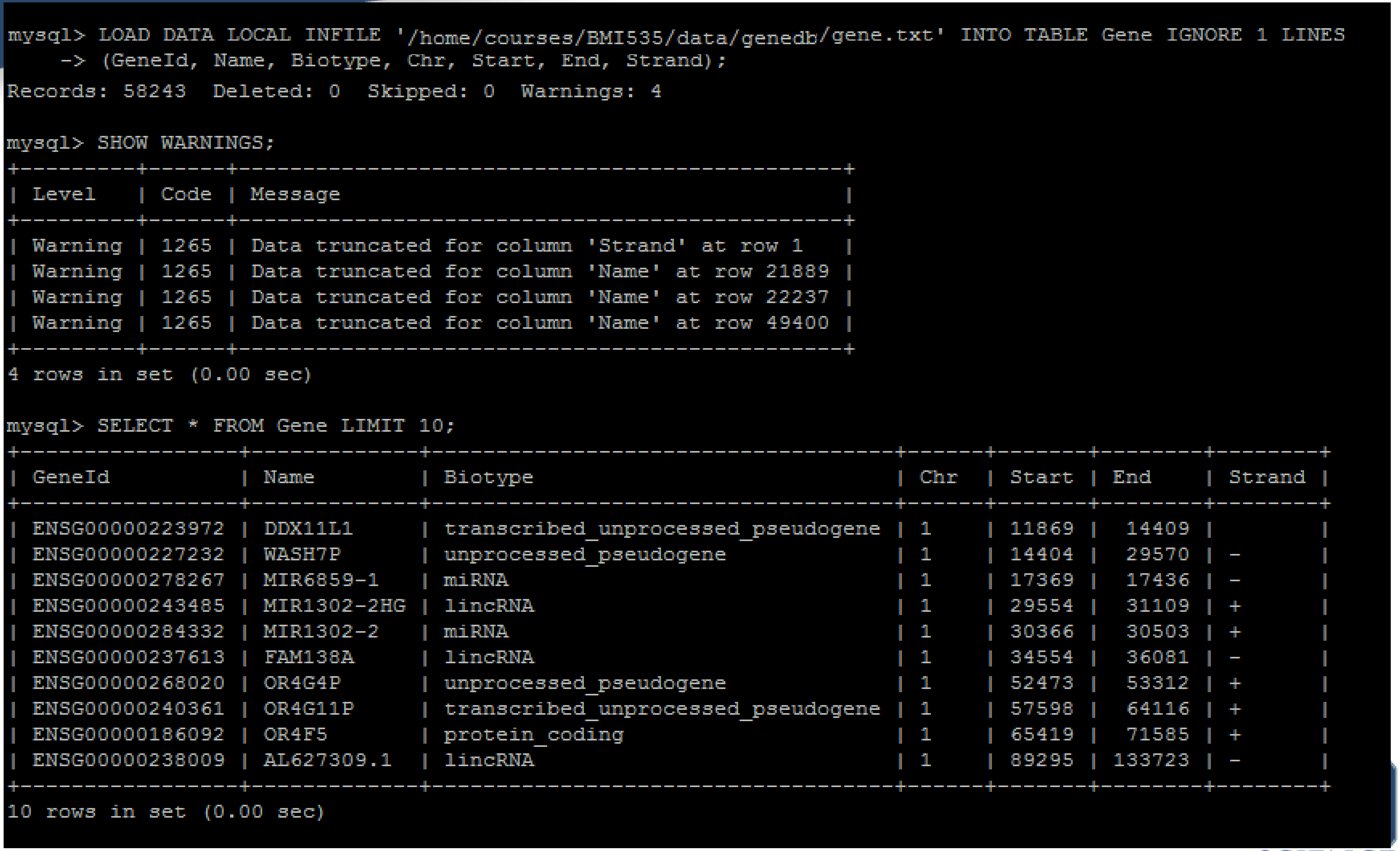
```
mysql> LOAD DATA LOCAL INFILE '/home/courses/BMI535/data/genedb/gene.txt' INTO TABLE Gene IGNORE 1 LINES
    -> (GeneId, Name, Biotype, Chr, Start, End, Strand);
Records: 58243  Deleted: 0  Skipped: 0  Warnings: 4
```

SQL: Loading Data From a File



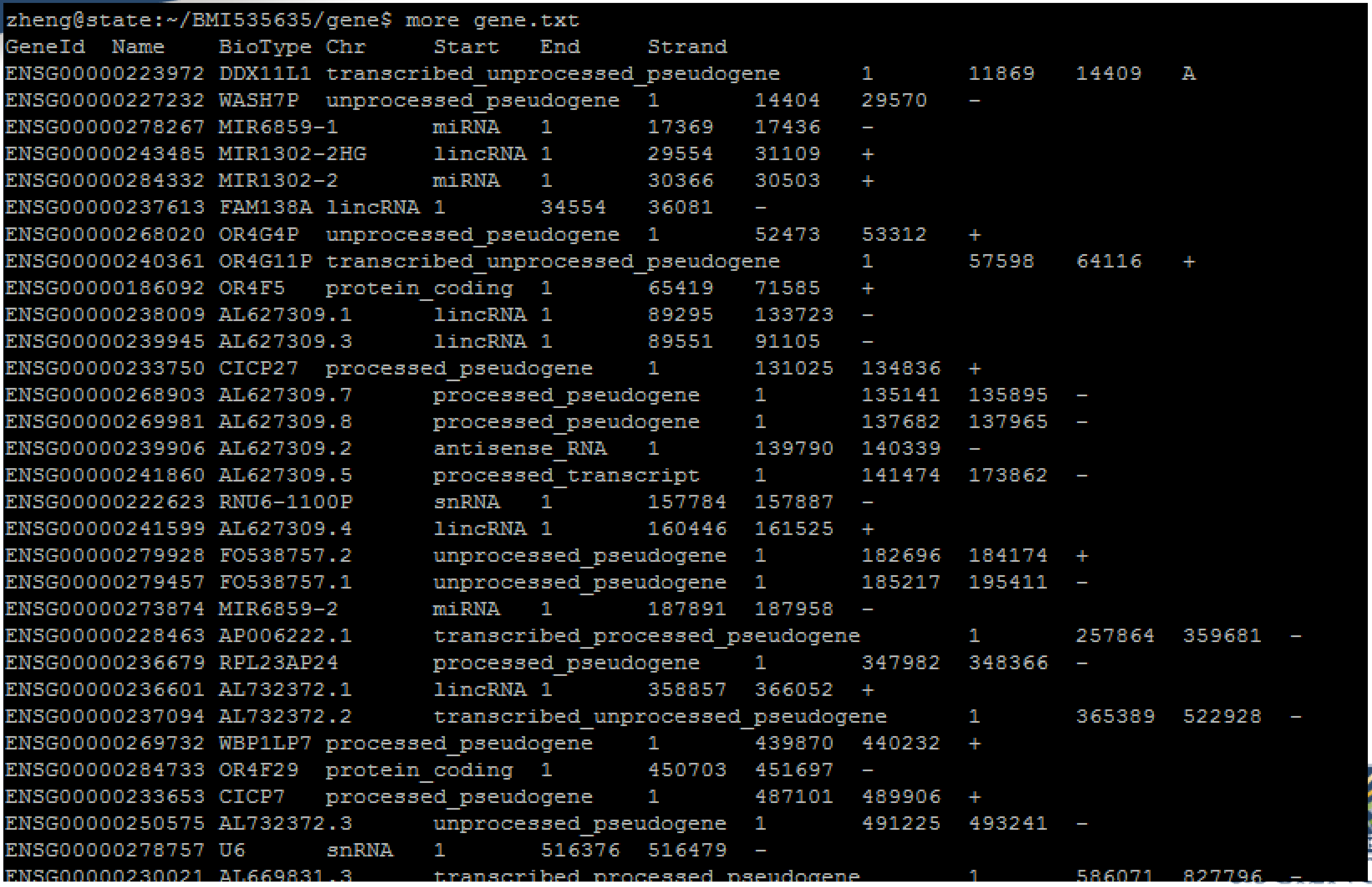
```
mysql> LOAD DATA LOCAL INFILE '/home/courses/BMI535/data/genedb/gene.txt' INTO TABLE Gene IGNORE 1 LINES
    -> (GeneId, Name, Biotype, Chr, Start, End, Strand);
Records: 58243  Deleted: 0  Skipped: 0  Warnings: 4
```

SQL: Loading Data From a File



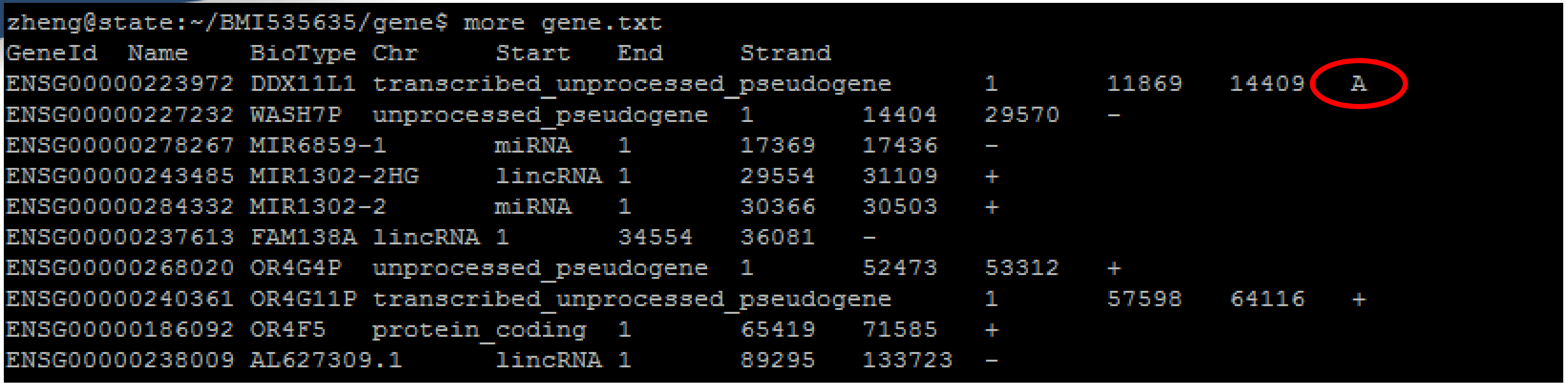
```
mysql> LOAD DATA LOCAL INFILE '/home/courses/BMI535/data/genedb/gene.txt' INTO TABLE Gene IGNORE 1 LINES
    -> (GeneId, Name, Biotype, Chr, Start, End, Strand);
Records: 58243  Deleted: 0  Skipped: 0  Warnings: 4

mysql> SHOW WARNINGS;
+---------+------+--------------------------------------------------+
| Level   | Code | Message                                          |
+---------+------+--------------------------------------------------+
| Warning | 1265 | Data truncated for column 'Strand' at row 1      |
| Warning | 1265 | Data truncated for column 'Name' at row 21889    |
| Warning | 1265 | Data truncated for column 'Name' at row 22237    |
| Warning | 1265 | Data truncated for column 'Name' at row 49400    |
+---------+------+--------------------------------------------------+
4 rows in set (0.00 sec)

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: Loading Data From a File

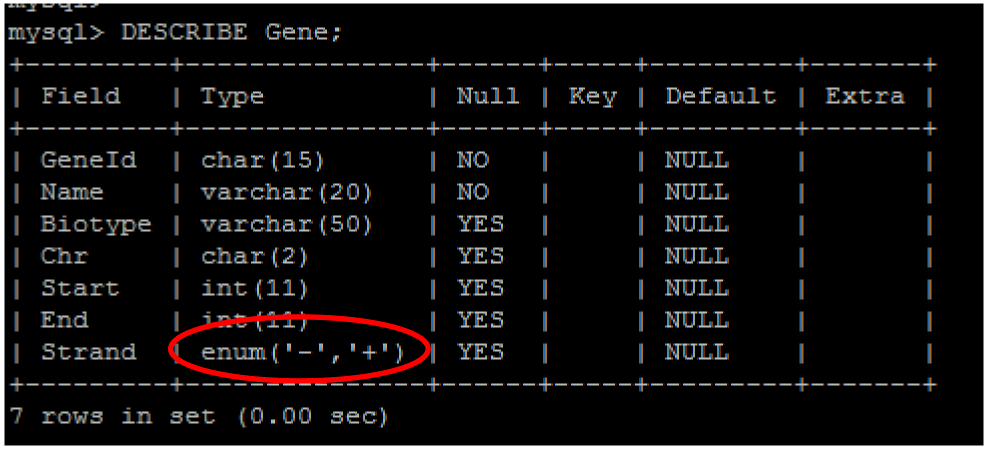


```
zheng@state:~/BMI535635/gene$ more gene.txt
GeneId  Name    BioType Chr     Start   End     Strand
ENSG00000223972 DDX11L1 transcribed_unprocessed_pseudogene      1       11869   14409   A
ENSG00000227232 WASH7P  unprocessed_pseudogene  1       14404   29570   -
ENSG00000278267 MIR6859-1       miRNA   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  -
ENSG00000239945 AL627309.3      lincRNA 1       89551   91105   -
ENSG00000233750 CICP27  processed_pseudogene    1       131025  134836  +
ENSG00000268903 AL627309.7      processed_pseudogene    1       135141  135895  -
ENSG00000269981 AL627309.8      processed_pseudogene    1       137682  137965  -
ENSG00000239906 AL627309.2      antisense_RNA   1       139790  140339  -
ENSG00000241860 AL627309.5      processed_transcript    1       141474  173862  -
ENSG00000222623 RNU6-1100P      snRNA   1       157784  157887  -
ENSG00000241599 AL627309.4      lincRNA 1       160446  161525  +
ENSG00000279928 FO538757.2      unprocessed_pseudogene  1       182696  184174  +
ENSG00000279457 FO538757.1      unprocessed_pseudogene  1       185217  195411  -
ENSG00000273874 MIR6859-2       miRNA   1       187891  187958  -
ENSG00000228463 AP006222.1      transcribed_processed_pseudogene        1       257864  359681  -
ENSG00000236679 RPL23AP24       processed_pseudogene    1       347982  348366  -
ENSG00000236601 AL732372.1      lincRNA 1       358857  366052  +
ENSG00000237094 AL732372.2      transcribed_unprocessed_pseudogene      1       365389  522928  -
ENSG00000269732 WBP1LP7 processed_pseudogene    1       439870  440232  +
ENSG00000284733 OR4F29  protein_coding  1       450703  451697  -
ENSG00000233653 CICP7   processed_pseudogene    1       487101  489906  +
ENSG00000250575 AL732372.3      unprocessed_pseudogene  1       491225  493241  -
ENSG00000278757 U6      snRNA   1       516376  516479  -
ENSG00000230021 AL669831.3      transcribed_processed_pseudogene        1       586071  827796  -
```

SQL: Loading Data From a File

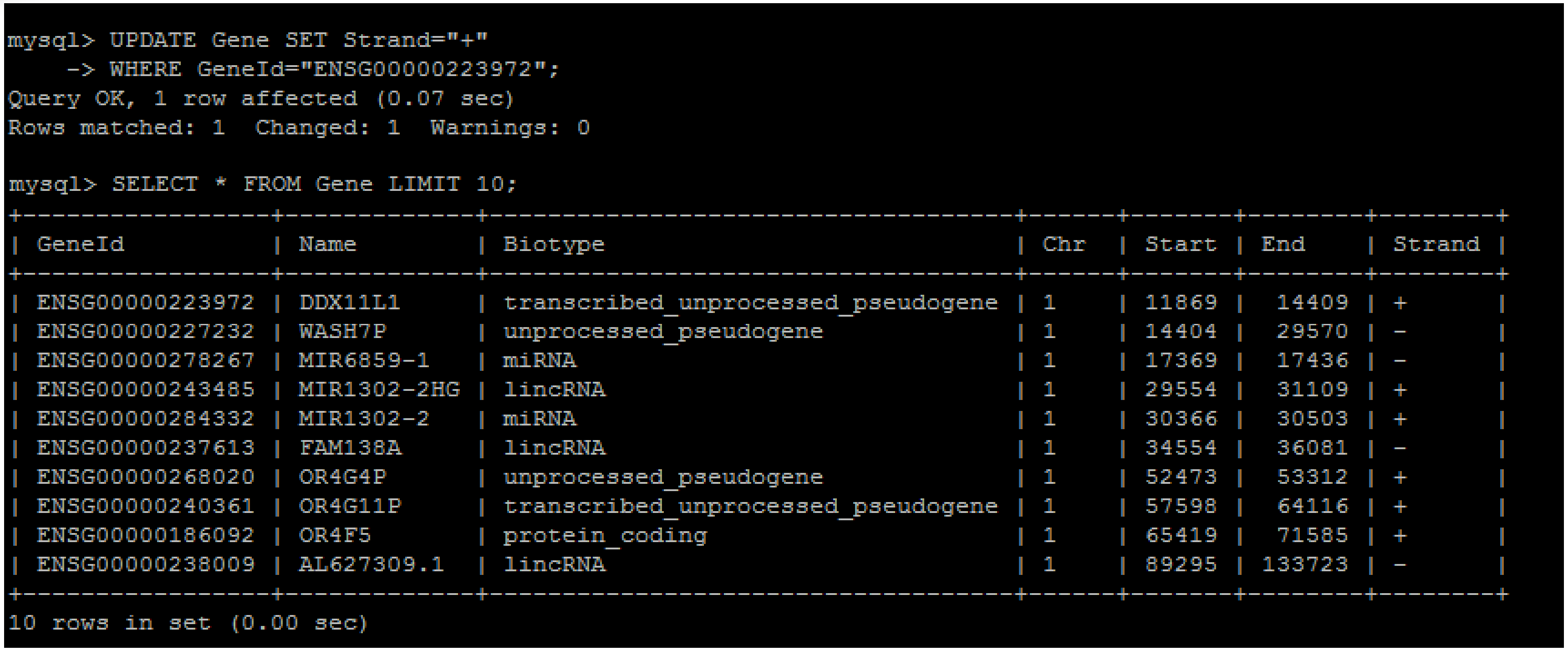


```
zheng@state:~/BMI535635/gene$ more gene.txt
GeneId  Name    BioType Chr     Start   End     Strand
ENSG00000223972 DDX11L1 transcribed_unprocessed_pseudogene      1       11869   14409   A
ENSG00000227232 WASH7P  unprocessed_pseudogene  1       14404   29570   -
ENSG00000278267 MIR6859-1       miRNA   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  -
```



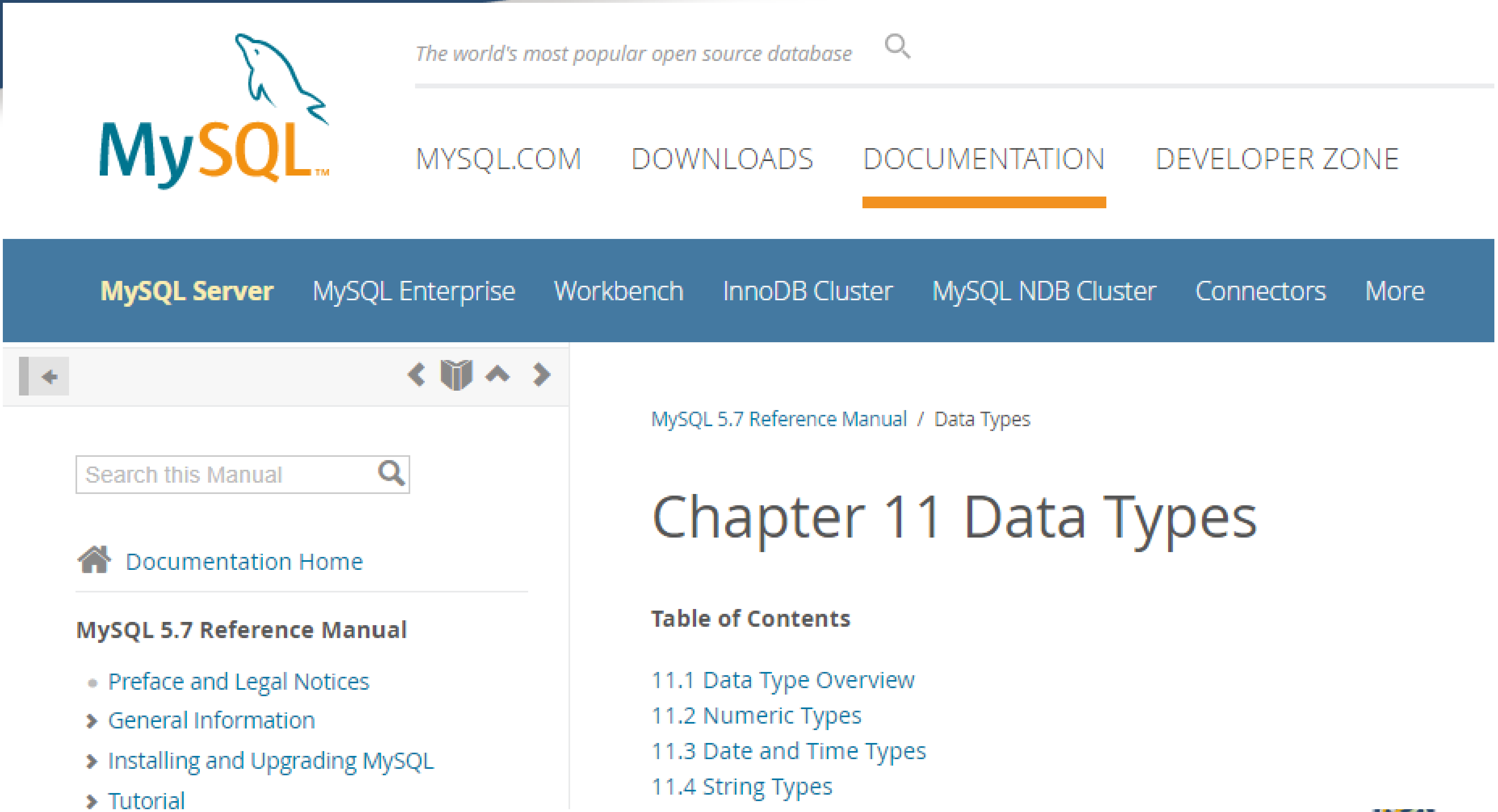
```
mysql>
mysql> DESCRIBE Gene;
+---------+---------------+------+-----+---------+-------+
| Field   | Type          | Null | Key | Default | Extra |
+---------+---------------+------+-----+---------+-------+
| GeneId  | char(15)      | NO   |     | NULL    |       |
| Name    | varchar(20)   | NO   |     | 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)
```

SQL: UPDATE



```
mysql> UPDATE Gene SET Strand="+"
    -> WHERE GeneId="ENSG00000223972";
Query OK, 1 row affected (0.07 sec)
Rows matched: 1  Changed: 1  Warnings: 0

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: Data Types


The world's most popular open source database

MYSQL.COM DOWNLOADS DOCUMENTATION DEVELOPER ZONE

MySQL Server MySQL Enterprise Workbench InnoDB Cluster MySQL NDB Cluster Connectors More

Search this Manual

Documentation Home

MySQL 5.7 Reference Manual

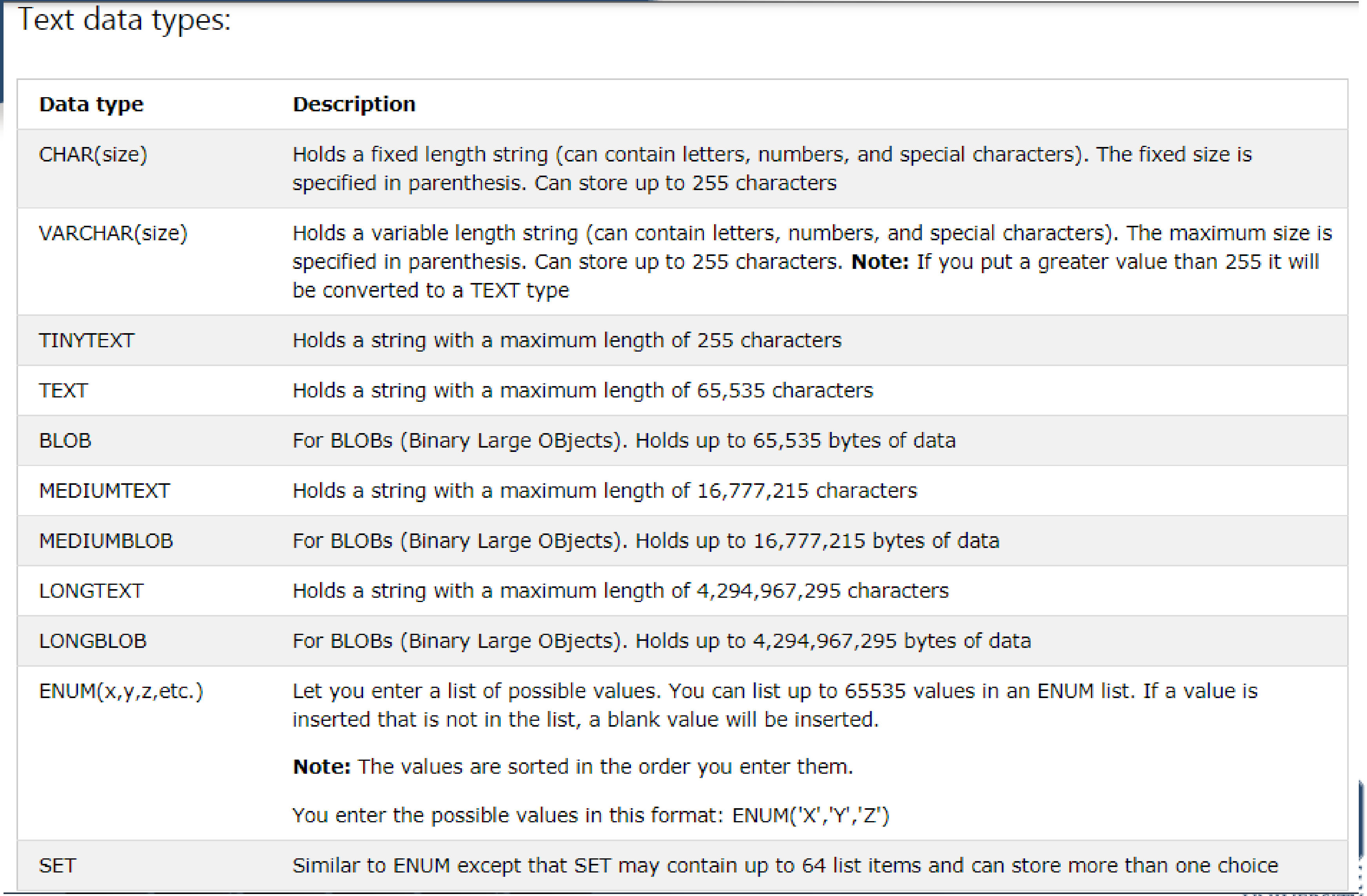
- Preface and Legal Notices
- General Information
- Installing and Upgrading MySQL
- Tutorial

MySQL 5.7 Reference Manual / Data Types

Chapter 11 Data Types

Table of Contents

SQL: Data Types

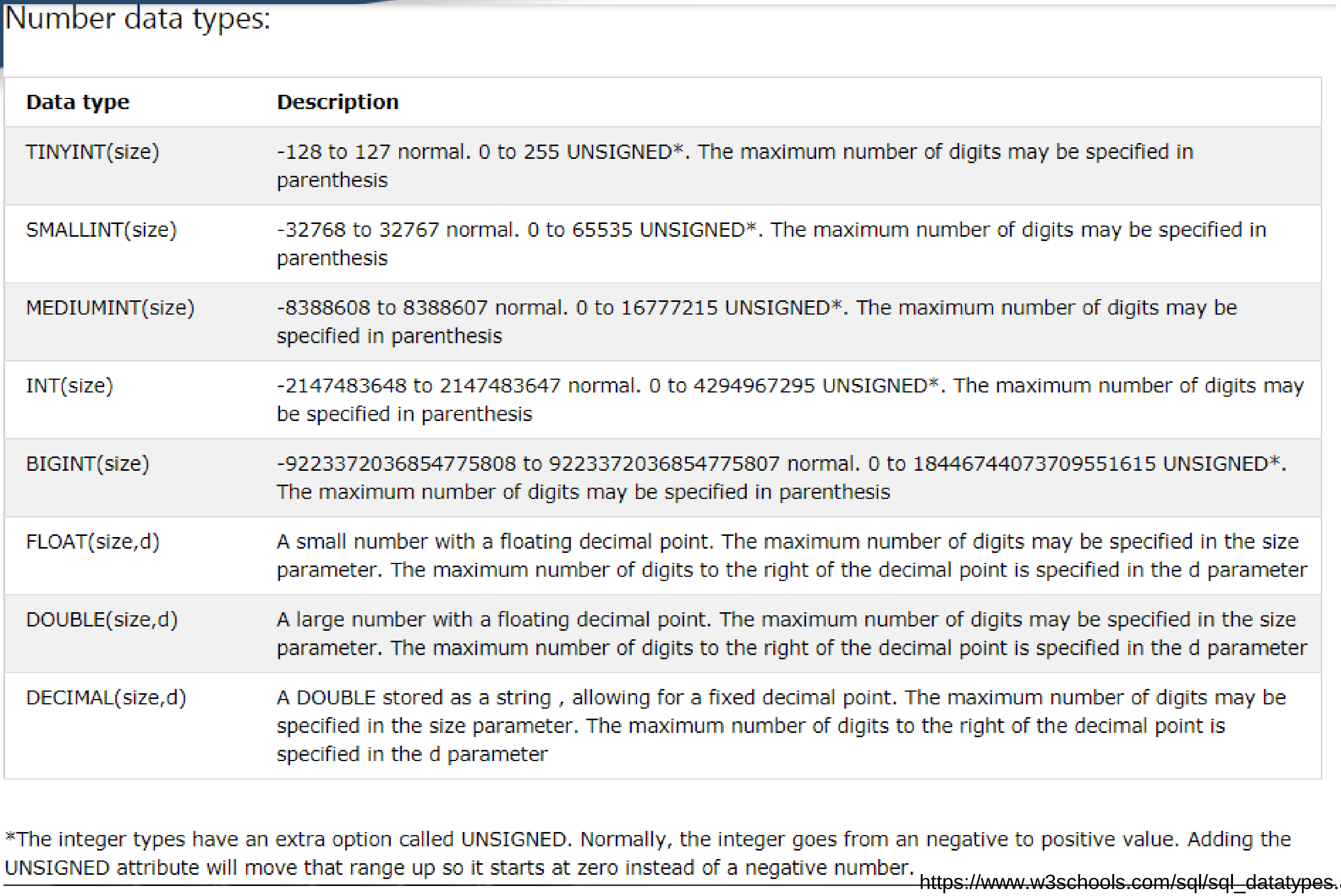


Text data types:

Data type	Description
CHAR(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters
VARCHAR(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters. **Note:** If you put a greater value than 255 it will be converted to a TEXT type
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT	Holds a string with a maximum length of 65,535 characters
BLOB	For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LONGBLOB	For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data
ENUM(x,y,z,etc.)	Let you enter a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted. **Note:** The values are sorted in the order you enter them. You enter the possible values in this format: ENUM('X','Y','Z')
SET	Similar to ENUM except that SET may contain up to 64 list items and can store more than one choice

https://www.w3schools.com/sql/sql_datatypes.asp

SQL: Data Types



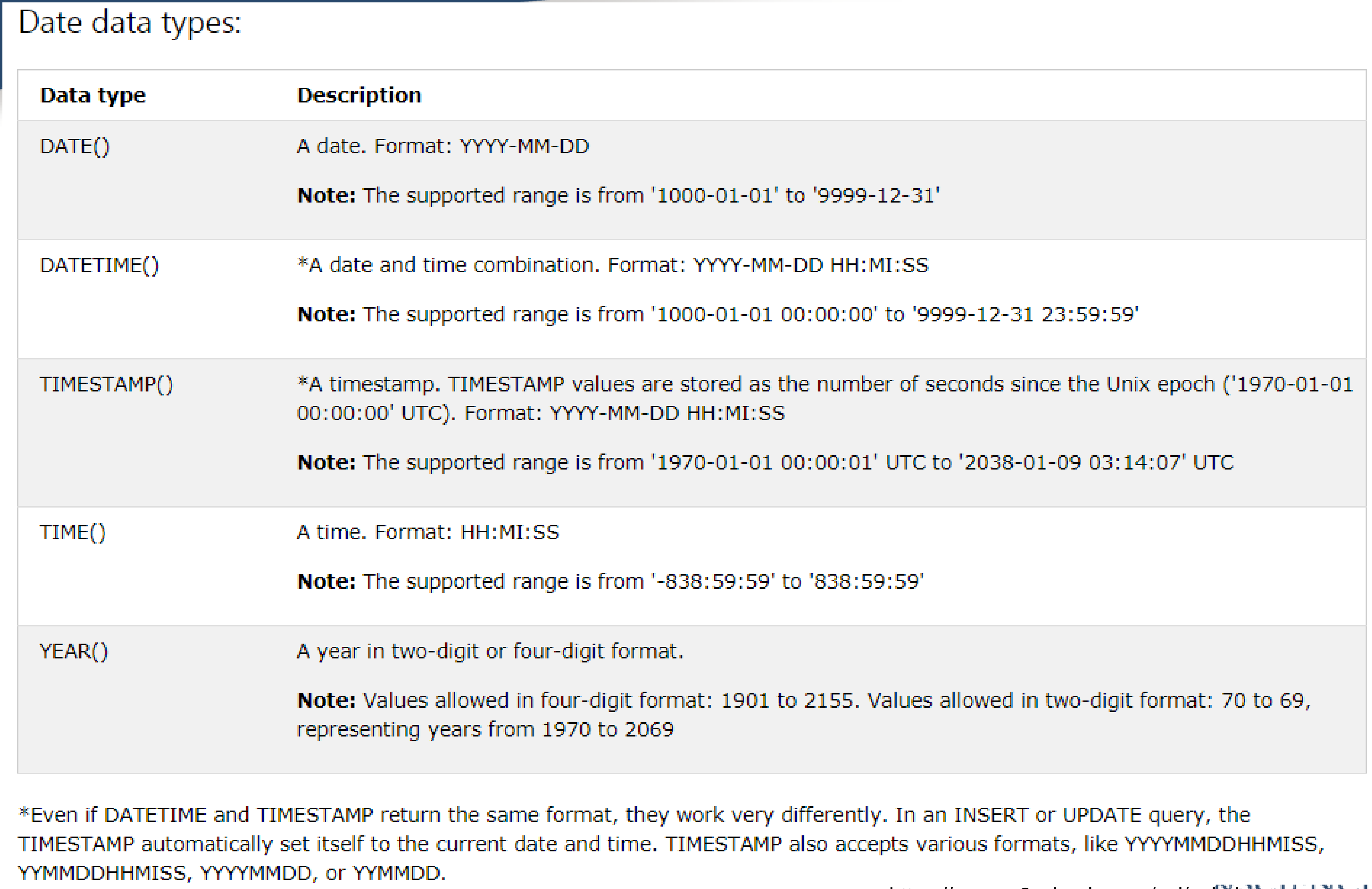
Number data types:

Data type	Description
TINYINT(size)	-128 to 127 normal. 0 to 255 UNSIGNED*. The maximum number of digits may be specified in parenthesis
SMALLINT(size)	-32768 to 32767 normal. 0 to 65535 UNSIGNED*. The maximum number of digits may be specified in parenthesis
MEDIUMINT(size)	-8388608 to 8388607 normal. 0 to 16777215 UNSIGNED*. The maximum number of digits may be specified in parenthesis
INT(size)	-2147483648 to 2147483647 normal. 0 to 4294967295 UNSIGNED*. The maximum number of digits may be specified in parenthesis
BIGINT(size)	-9223372036854775808 to 9223372036854775807 normal. 0 to 18446744073709551615 UNSIGNED*. The maximum number of digits may be specified in parenthesis
FLOAT(size,d)	A small number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter
DOUBLE(size,d)	A large number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter
DECIMAL(size,d)	A DOUBLE stored as a string , allowing for a fixed decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter

*The integer types have an extra option called UNSIGNED. Normally, the integer goes from an negative to positive value. Adding the UNSIGNED attribute will move that range up so it starts at zero instead of a negative number.

https://www.w3schools.com/sql/sql_datatypes.asp

SQL: Data Types



Date data types:

Data type	Description
DATE()	A date. Format: YYYY-MM-DD **Note:** The supported range is from '1000-01-01' to '9999-12-31'
DATETIME()	*A date and time combination. Format: YYYY-MM-DD HH:MI:SS **Note:** The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'
TIMESTAMP()	*A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD HH:MI:SS **Note:** The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC
TIME()	A time. Format: HH:MI:SS **Note:** The supported range is from '-838:59:59' to '838:59:59'
YEAR()	A year in two-digit or four-digit format. **Note:** Values allowed in four-digit format: 1901 to 2155. Values allowed in two-digit format: 70 to 69, representing years from 1970 to 2069

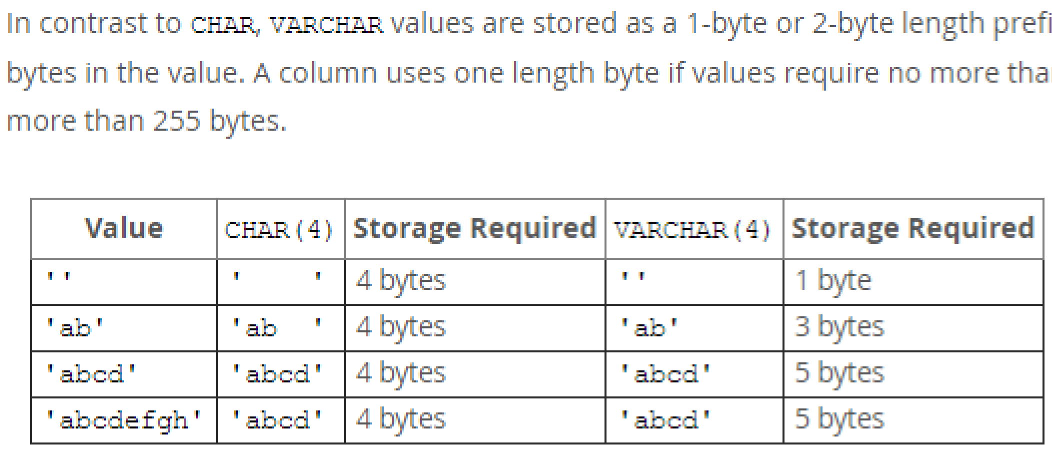
*Even if DATETIME and TIMESTAMP return the same format, they work very differently. In an INSERT or UPDATE query, the TIMESTAMP automatically set itself to the current date and time. TIMESTAMP also accepts various formats, like YYYYMMDDHHMISS, YYMMDDHHMISS, YYYYMMDD, or YYMMDD.

https://www.w3schools.com/sql/sql_datatypes.asp

SQL: Data Types

Data Types:

- enforces data types, ranges (integrity constraints)
- aids in efficient storage of data



In contrast to `CHAR`, `VARCHAR` values are stored as a 1-byte or 2-byte length prefix plus data. The length prefix indicates the number of bytes in the value. A column uses one length byte if values require no more than 255 bytes, two length bytes if values may require more than 255 bytes.

Value	`CHAR(4)`	Storage Required	`VARCHAR(4)`	Storage Required
`''`	`'    '`	4 bytes	`''`	1 byte
`'ab'`	`'ab  '`	4 bytes	`'ab'`	3 bytes
`'abcd'`	`'abcd'`	4 bytes	`'abcd'`	5 bytes
`'abcdefgh'`	`'abcd'`	4 bytes	`'abcd'`	5 bytes

https://dev.mysql.com/doc/refman/5.7/en/char.html

Integrity Constraints

Conditions that must be true for any instance of the database

- specified when schema is defined
- checked when relations are added/modified
- DBMS does not allow illegal instances
- avoids data entry errors

Key Constraints

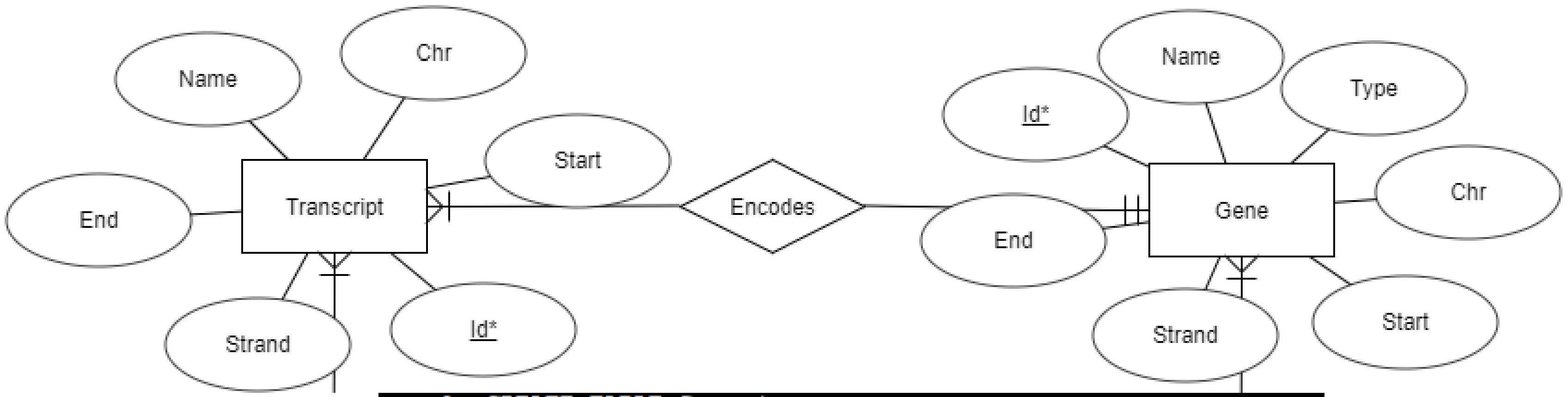
Candidate Key

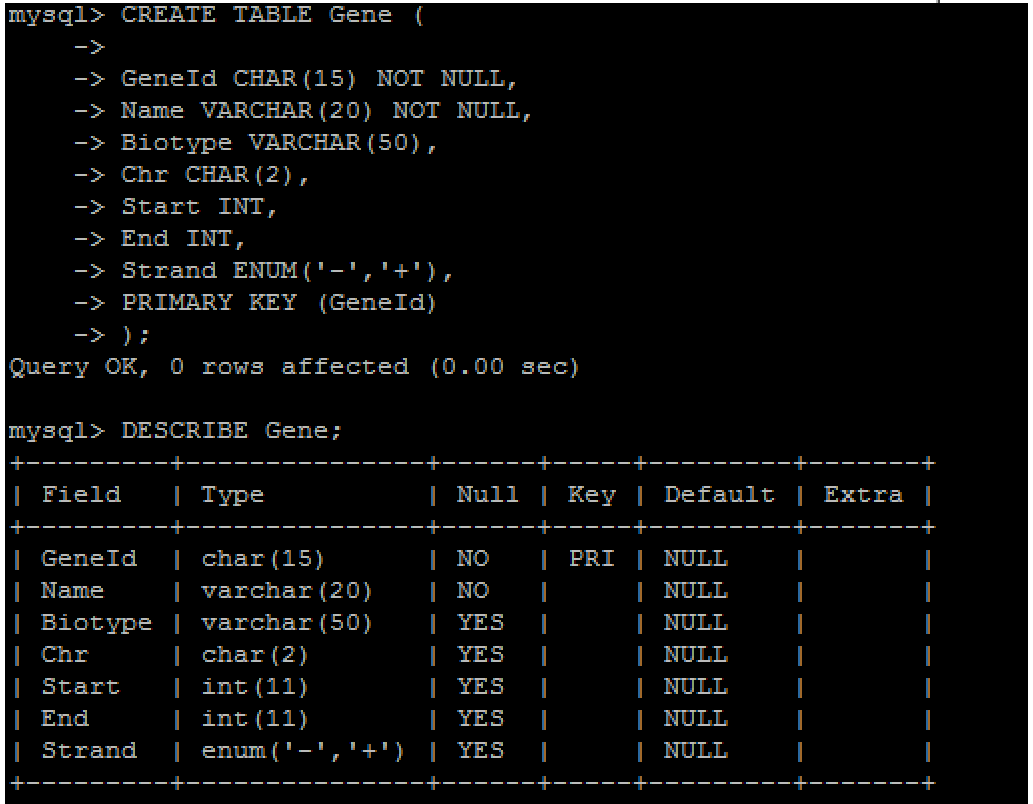
- column/set of columns that uniquely identifies a row
- unique gene identifier, unique transcript identifier

Primary Key

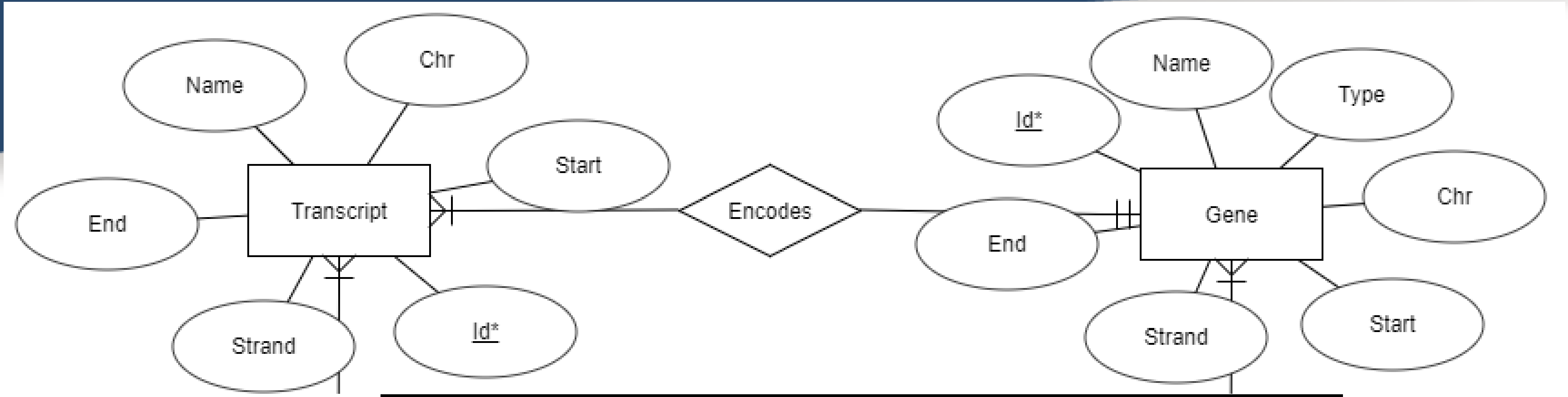
- logical unique identifier
- querying, data organization

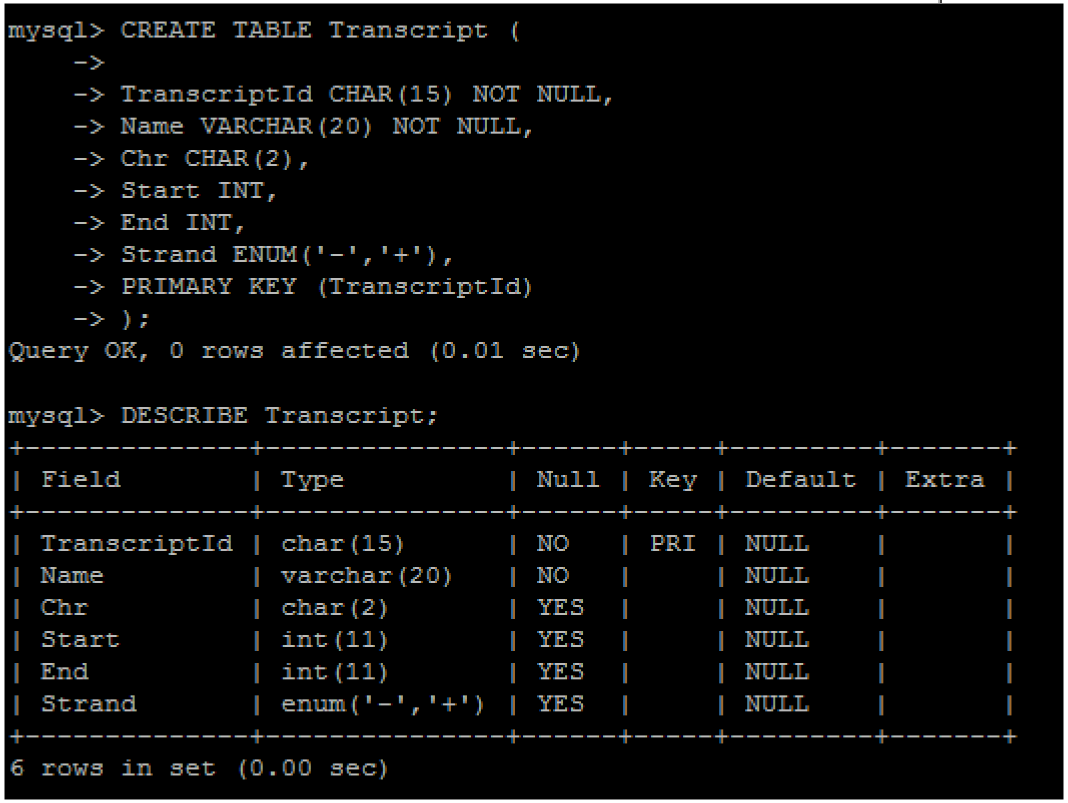
Primary Key




```
mysql> CREATE TABLE Gene (
    ->
    -> GeneId CHAR(15) NOT NULL,
    -> Name VARCHAR(20) NOT NULL,
    -> Biotype VARCHAR(50),
    -> Chr CHAR(2),
    -> Start INT,
    -> End INT,
    -> Strand ENUM('-','+'),
    -> PRIMARY KEY (GeneId)
    -> );
Query OK, 0 rows affected (0.00 sec)

mysql> DESCRIBE Gene;
+---------+---------------+------+-----+---------+-------+
| Field   | Type          | Null | Key | Default | Extra |
+---------+---------------+------+-----+---------+-------+
| GeneId  | char(15)      | NO   | PRI | 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    |       |
+---------+---------------+------+-----+---------+-------+
```

Primary Key



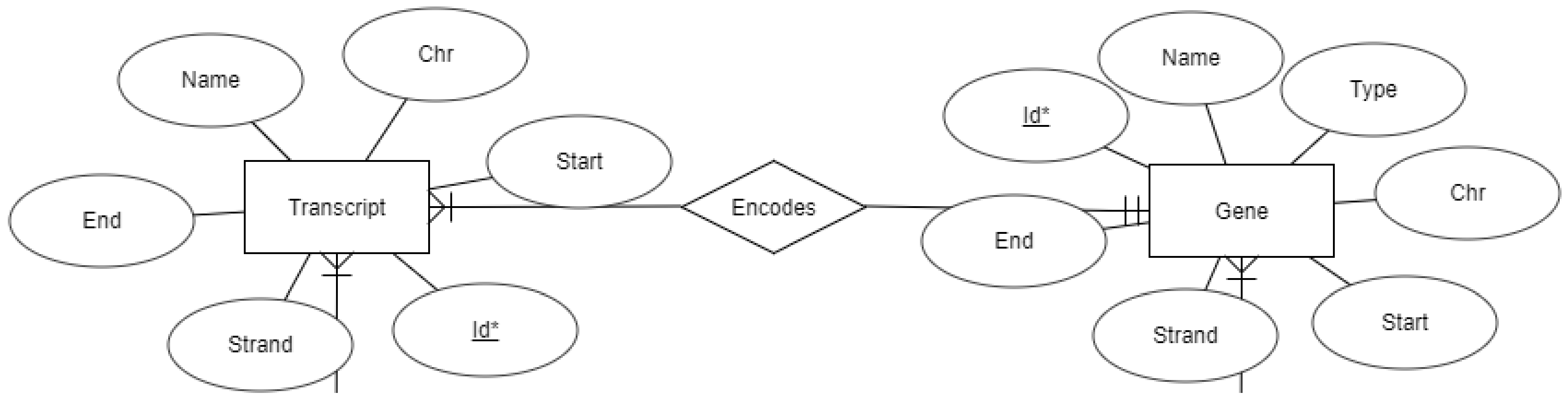


```
mysql> CREATE TABLE Transcript (
    ->
    -> TranscriptId CHAR(15) NOT NULL,
    -> Name VARCHAR(20) NOT NULL,
    -> Chr CHAR(2),
    -> Start INT,
    -> End INT,
    -> Strand ENUM('-','+'),
    -> PRIMARY KEY (TranscriptId)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> DESCRIBE Transcript;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| TranscriptId | char(15)     | NO   | PRI | NULL    |       |
| Name         | varchar(20)  | NO   |     | NULL    |       |
| Chr          | char(2)      | YES  |     | NULL    |       |
| Start        | int(11)      | YES  |     | NULL    |       |
| End          | int(11)      | YES  |     | NULL    |       |
| Strand       | enum('-','+') | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

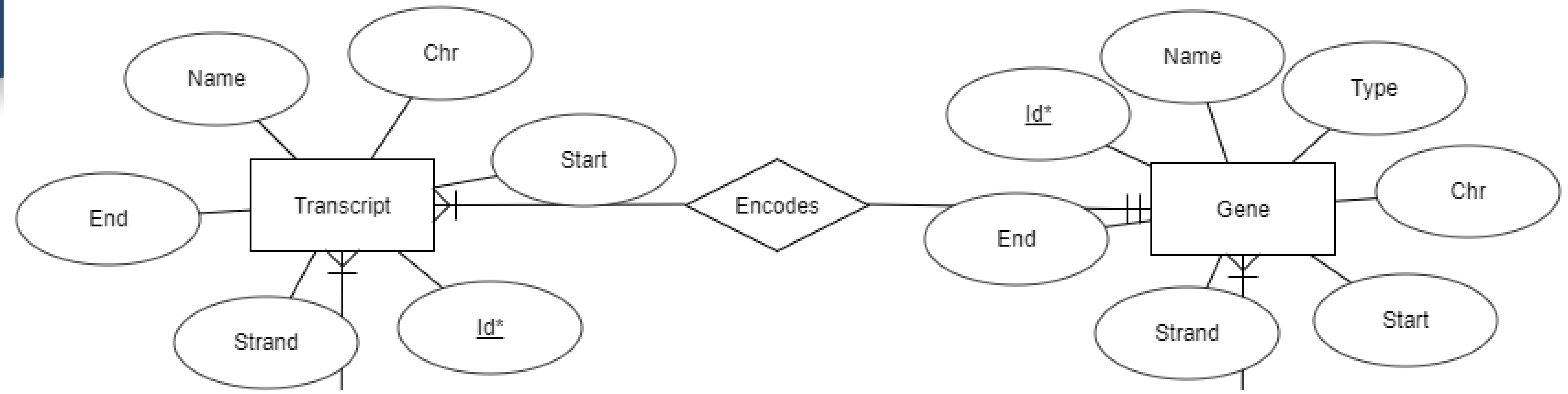
Referential Integrity

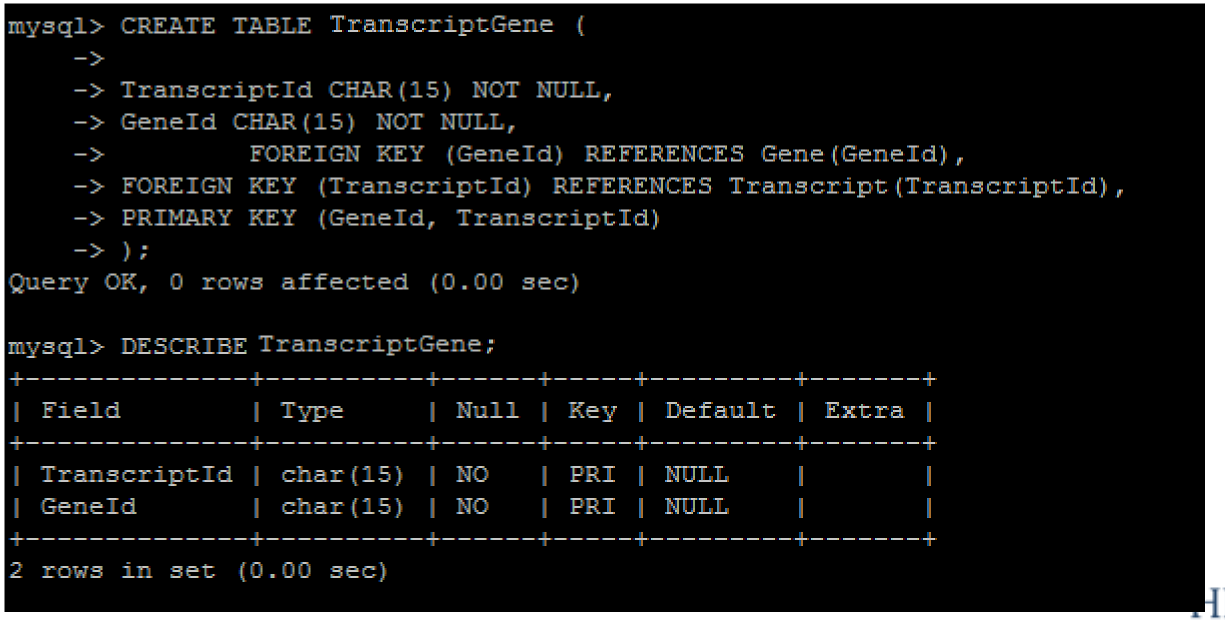
Information stored in one table linked to information stored in another

- modification needs a check/modification of the other
    - only transcripts with a valid gene is entered
    - every transcript must reference a gene



Foreign Key

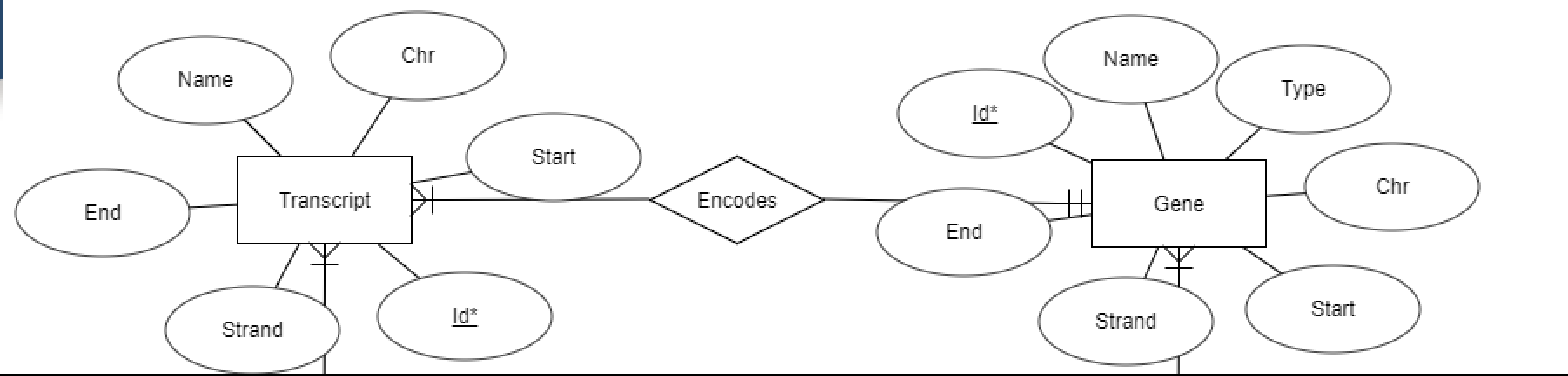


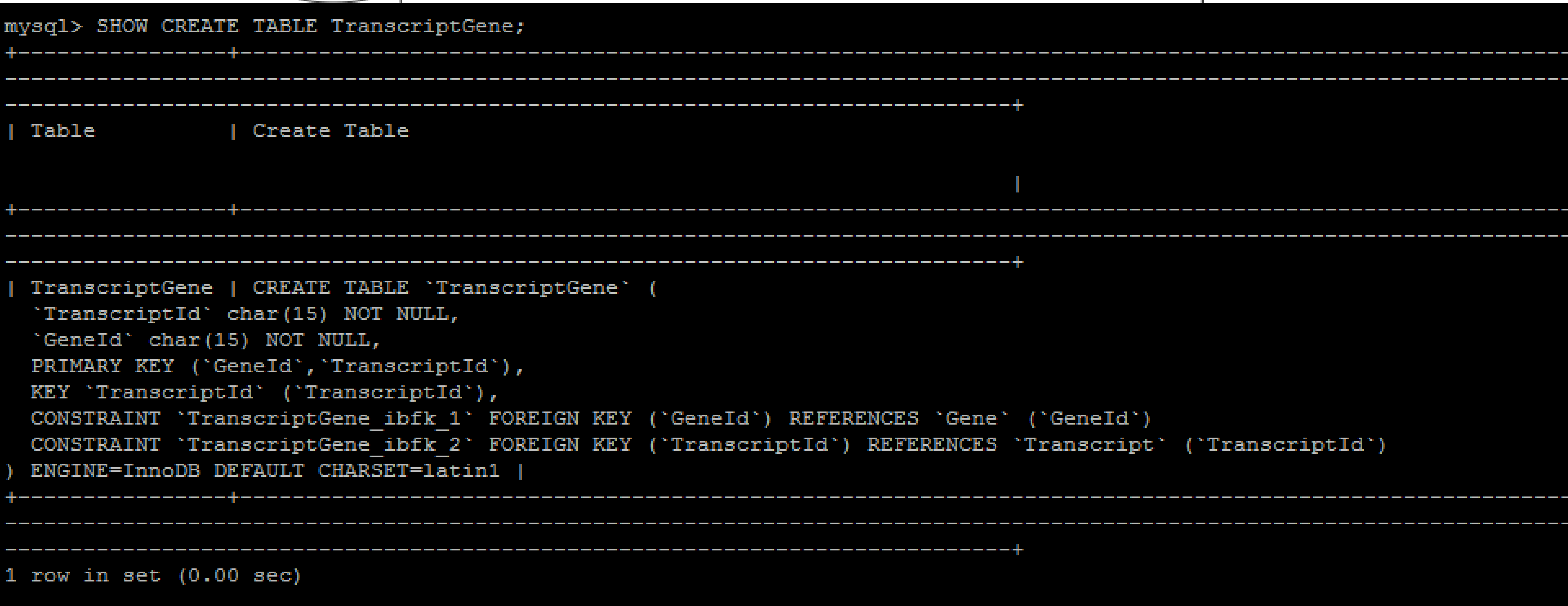


```
mysql> CREATE TABLE TranscriptGene (
    ->
    -> TranscriptId CHAR(15) NOT NULL,
    -> GeneId CHAR(15) NOT NULL,
    ->           FOREIGN KEY (GeneId) REFERENCES Gene(GeneId),
    -> FOREIGN KEY (TranscriptId) REFERENCES Transcript(TranscriptId),
    -> PRIMARY KEY (GeneId, TranscriptId)
    -> );
Query OK, 0 rows affected (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)
```

*Foreign key must match primary key of other table

Foreign Key

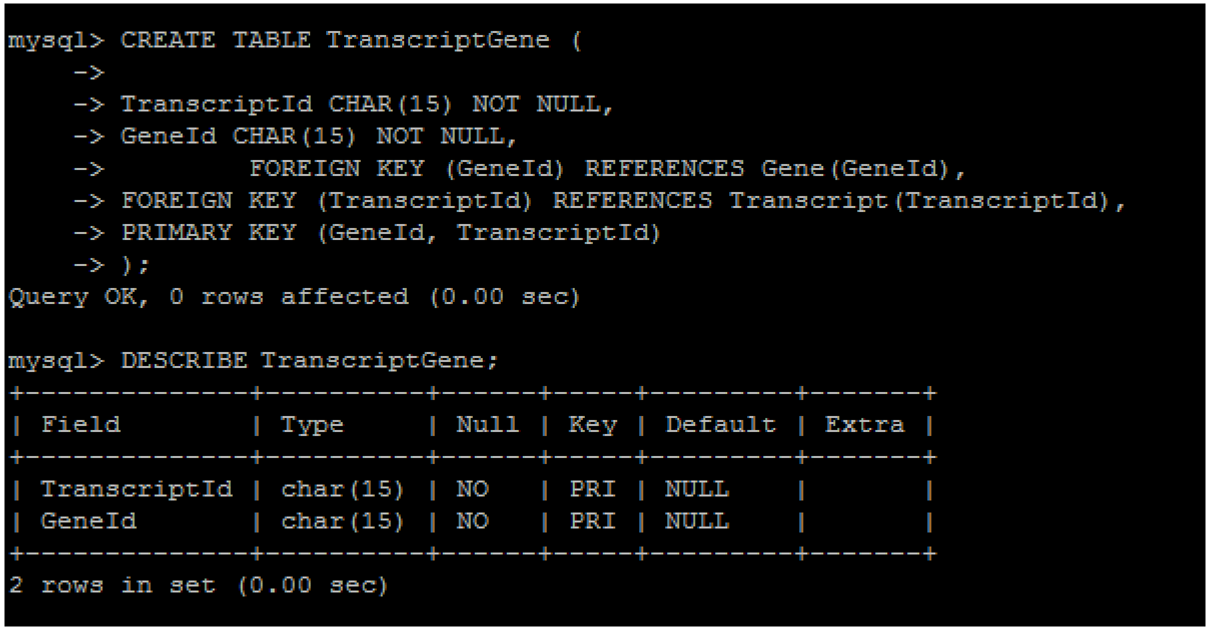




```
mysql> SHOW CREATE TABLE TranscriptGene;
+----------------+--------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------+
| Table          | Create Table


                                                                           |
+----------------+--------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------+
| TranscriptGene | CREATE TABLE `TranscriptGene` (
  `TranscriptId` char(15) NOT NULL,
  `GeneId` char(15) NOT NULL,
  PRIMARY KEY (`GeneId`,`TranscriptId`),
  KEY `TranscriptId` (`TranscriptId`),
  CONSTRAINT `TranscriptGene_ibfk_1` FOREIGN KEY (`GeneId`) REFERENCES `Gene` (`GeneId`)
  CONSTRAINT `TranscriptGene_ibfk_2` FOREIGN KEY (`TranscriptId`) REFERENCES `Transcript` (`TranscriptId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+----------------+--------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

Enforcing Referential Integrity

What should happen if you insert a transcript with a GeneId not found in the Gene table?



```
mysql> CREATE TABLE TranscriptGene (
    ->
    -> TranscriptId CHAR(15) NOT NULL,
    -> GeneId CHAR(15) NOT NULL,
    ->         FOREIGN KEY (GeneId) REFERENCES Gene(GeneId),
    -> FOREIGN KEY (TranscriptId) REFERENCES Transcript(TranscriptId),
    -> PRIMARY KEY (GeneId, TranscriptId)
    -> );
Query OK, 0 rows affected (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)
```

Enforcing Referential Integrity

What should happen if you insert a transcript with a GeneId not found in the Gene table?

What should happen if you insert a gene into the Gene table that is not referenced by a transcript?

Enforcing Referential Integrity

What should happen if you insert a transcript with a GeneId not found in the Gene table?

What should happen if you insert a gene into the Gene table that is not referenced by a transcript?

What should happen if a gene in the Gene table is deleted but it is referenced by a transcript?

Enforcing Referential Integrity

What should happen if you insert a transcript with a GeneId not found in the Gene table?

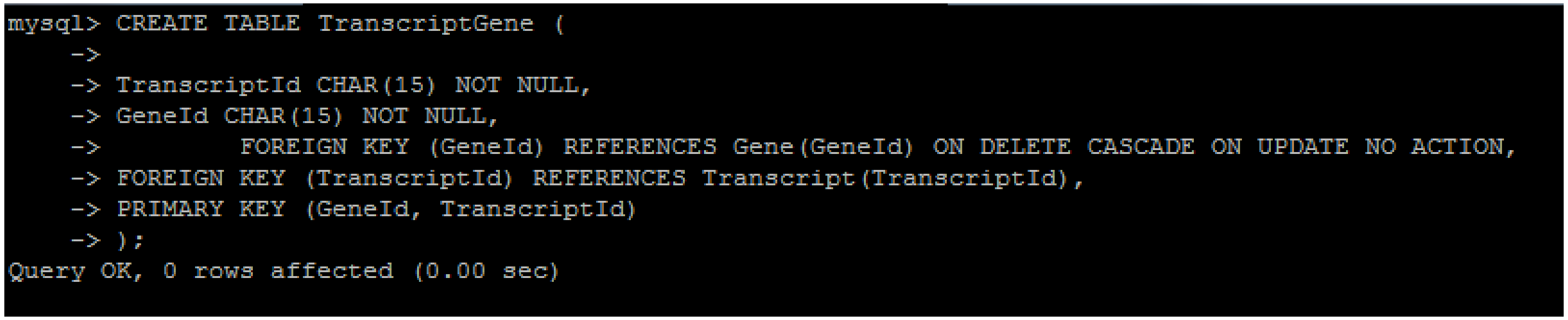
What should happen if you insert a gene into the Gene table that is not referenced in the Transcript table?

What should happen if a gene in the Gene table is deleted but it is referenced by a transcript?
- delete all associated transcripts?
- disallow deletion of the gene?

Enforcing Referential Integrity

SQL options

- no action: delete/update is rejected
- cascade: delete all referenced rows
- set null/set default: sets foreign key value of referencing row



```
mysql> CREATE TABLE TranscriptGene (
    ->
    -> TranscriptId CHAR(15) NOT NULL,
    -> GeneId CHAR(15) NOT NULL,
    ->         FOREIGN KEY (GeneId) REFERENCES Gene(GeneId) ON DELETE CASCADE ON UPDATE NO ACTION,
    -> FOREIGN KEY (TranscriptId) REFERENCES Transcript(TranscriptId),
    -> PRIMARY KEY (GeneId, TranscriptId)
    -> );
Query OK, 0 rows affected (0.00 sec)
```

Database Design Considerations

Requirements analysis

- type of data, types of queries, performance requirements
- data model

Conceptual Database Design

- high level description/constraints of the data (ER Model)

Logical Database Design

- convert conceptual design into a database schema

Conceptual Design

Gene(id:str, name:str, type:str, chr:str, start:int, end:int, strand:str)

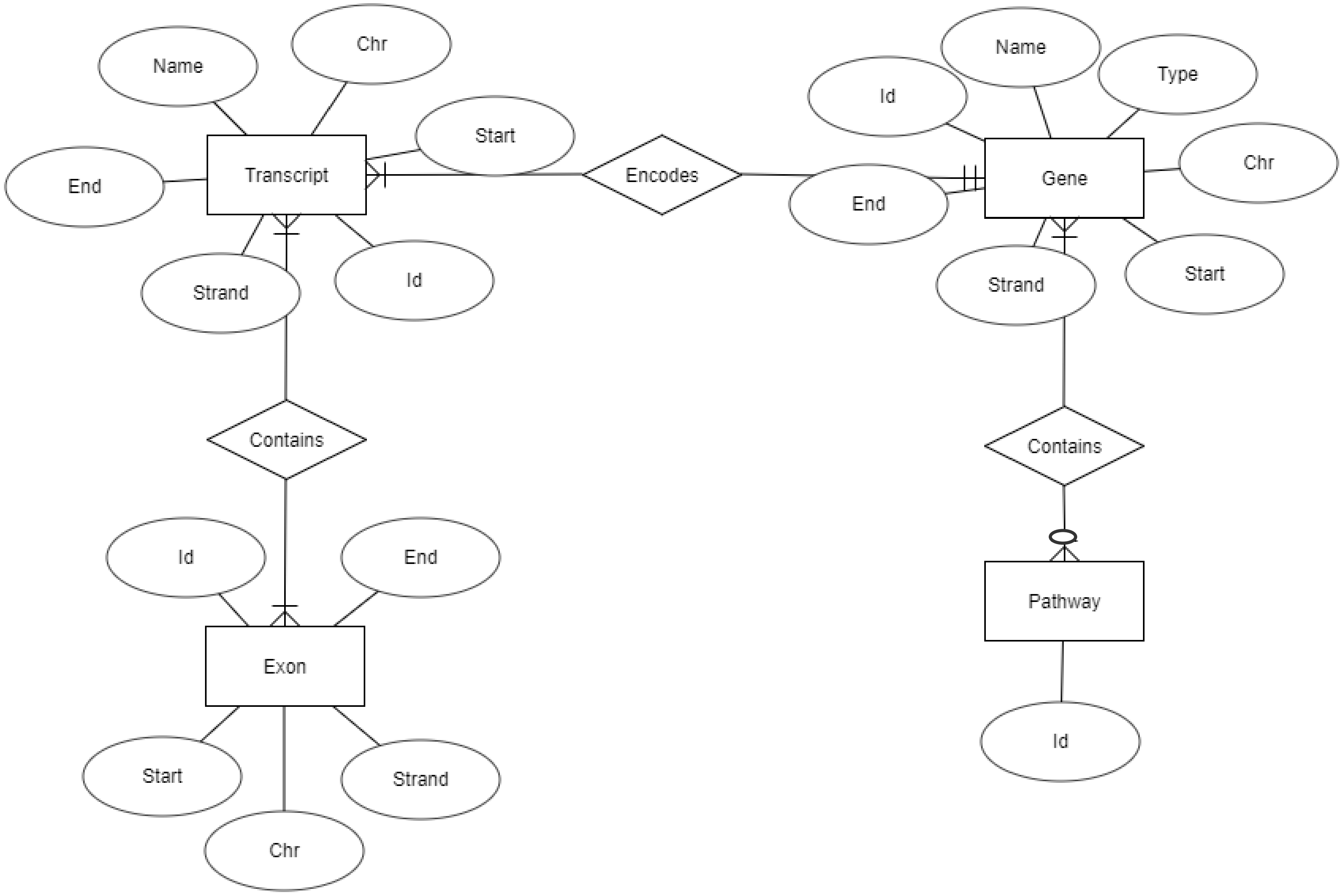
Transcript(id:str, name:str, chr:str, start:int, end:int, strand:str)

Exon(id:str, chr:str, start:int, end:int, strand:str)

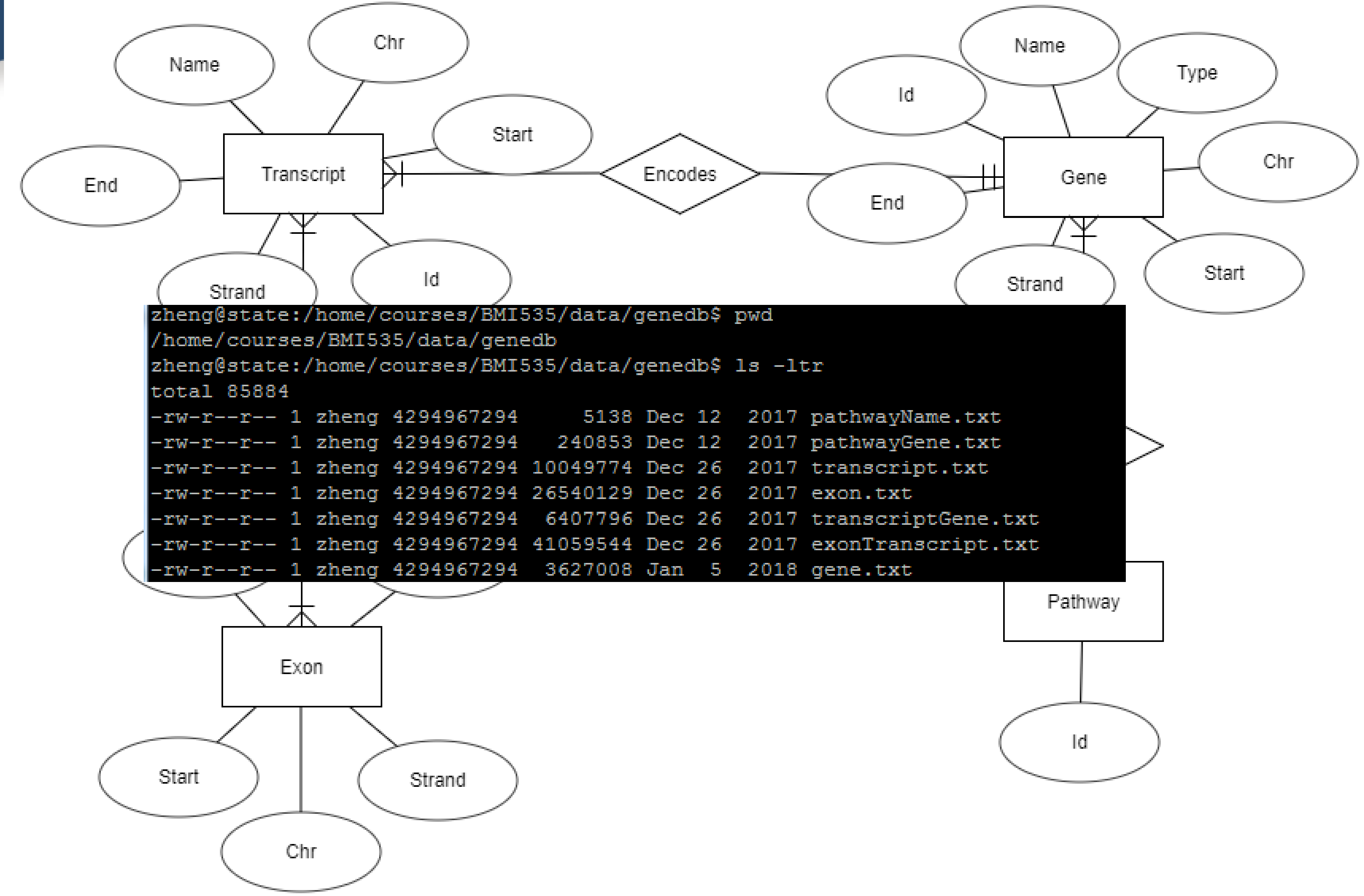
Pathway(id:str)

OREGON HEALTH & SCIENCE UNIVERSITY

ER Model


Name
Chr
Start
End
Transcript
Strand
Id
Encodes
Id
Name
Type
Chr
Gene
End
Strand
Start
Contains
Id
End
Exon
Start
Strand
Chr
Contains
Pathway
Id

ER Model



```
zheng@state:/home/courses/BMI535/data/genedb$ pwd
/home/courses/BMI535/data/genedb
zheng@state:/home/courses/BMI535/data/genedb$ ls -ltr
total 85884
-rw-r--r-- 1 zheng 4294967294     5138 Dec 12  2017 pathwayName.txt
-rw-r--r-- 1 zheng 4294967294   240853 Dec 12  2017 pathwayGene.txt
-rw-r--r-- 1 zheng 4294967294 10049774 Dec 26  2017 transcript.txt
-rw-r--r-- 1 zheng 4294967294 26540129 Dec 26  2017 exon.txt
-rw-r--r-- 1 zheng 4294967294  6407796 Dec 26  2017 transcriptGene.txt
-rw-r--r-- 1 zheng 4294967294 41059544 Dec 26  2017 exonTranscript.txt
-rw-r--r-- 1 zheng 4294967294  3627008 Jan  5  2018 gene.txt
```