SQL / Data Storage

cs380

GTowell, Sep 14

There are lots of ways to store data

- Considering only machine readable
  - Flat text files
    - fixed width fields
    - Delimited, e.g., CSV, ...
    - Advantages/disadvantages
- Binary formats
  - Java Objects, C bits
    - Advantages/Disadvantages
- Image-based formats:
  - QR
  - GeoTiff, ...

```
 27.08 50.15 98.72 57.62
 70.31 43.93 66.87 86.02
 72.18  7.41 87.81 92.04
 26.47 27.28 98.89 44.58
 14.66 37.28 93.62 61.84
 68.10 11.17 91.69 41.61
```

```
54.4,90.40,4.746,37.169050714
72.9,37.6,96.7205,51.5027943330
70.0,84.6,53.617,7.7
34.7,50.132,92.9177,17.07535291
79.9,11.06,38.20521,47.5
82.9,69.0,87.1345433,17.8180
```

```
try (ObjectOutputStream oos = new ObjectOutputStream(
       new FileOutputStream("AA.out"))) {
            oos.writeObject(als);
        } catch (Exception ee) {
            System.err.println(ee);
        }
```

```
FILE* f = fopen("wb.bin", “r");
char c[20];
r = fwrite(c, 20, sizeof(char), f);
```

Other data file formats

- Self documenting!!
  - netCDF(!)
- Well defined format, easily parsed (if not self documenting)
  - JSON(!)

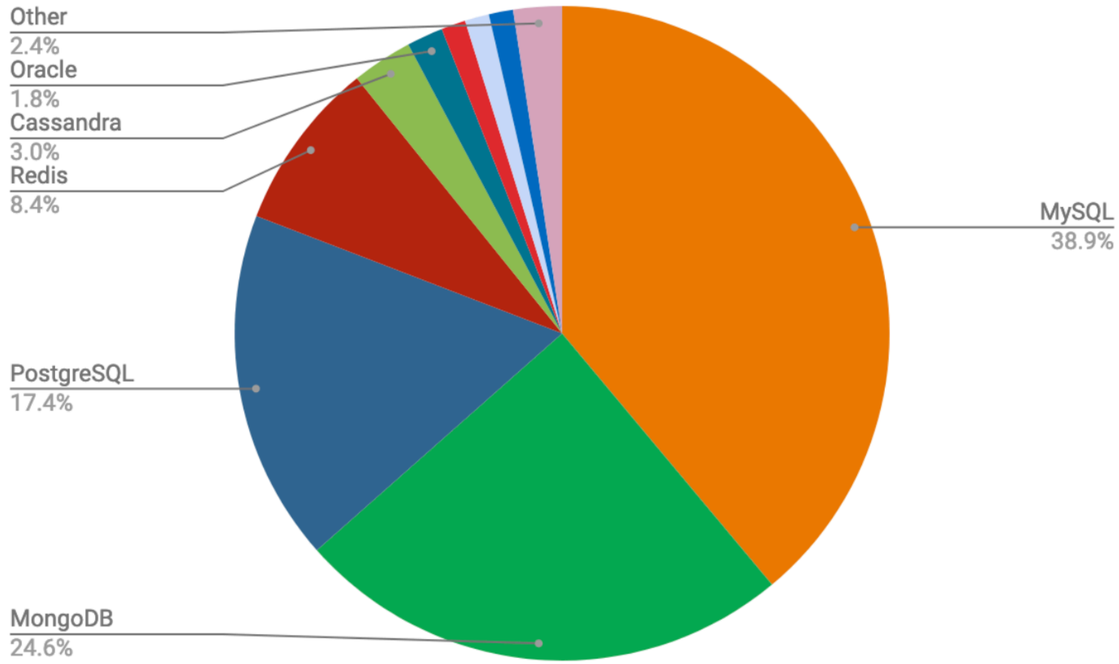
- XML
  - We will come back to this later in semester

- URLs
  - what are all of the parts of a URL?

Relational DataBases

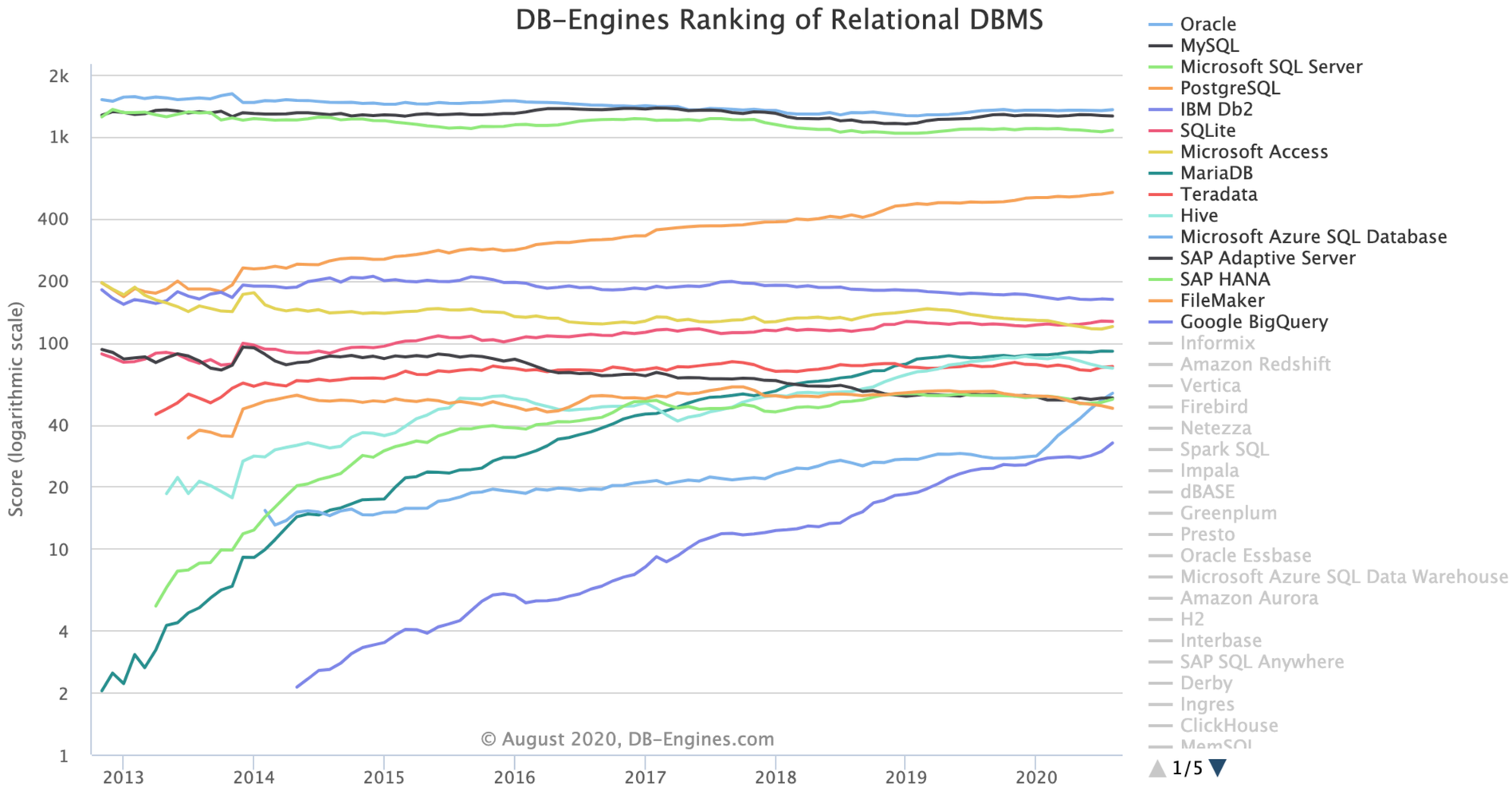
- With databases you are not reading/writing files, rather you are asking another process for information
  - Other examples of doing this
- Advantages/Disadvantages

- Relational database products
  - Open Source: MySql, Postgres,...
  - Commercial: Oracle, SQL Server, ...



“popularity”

Relational Databases ranked by usage

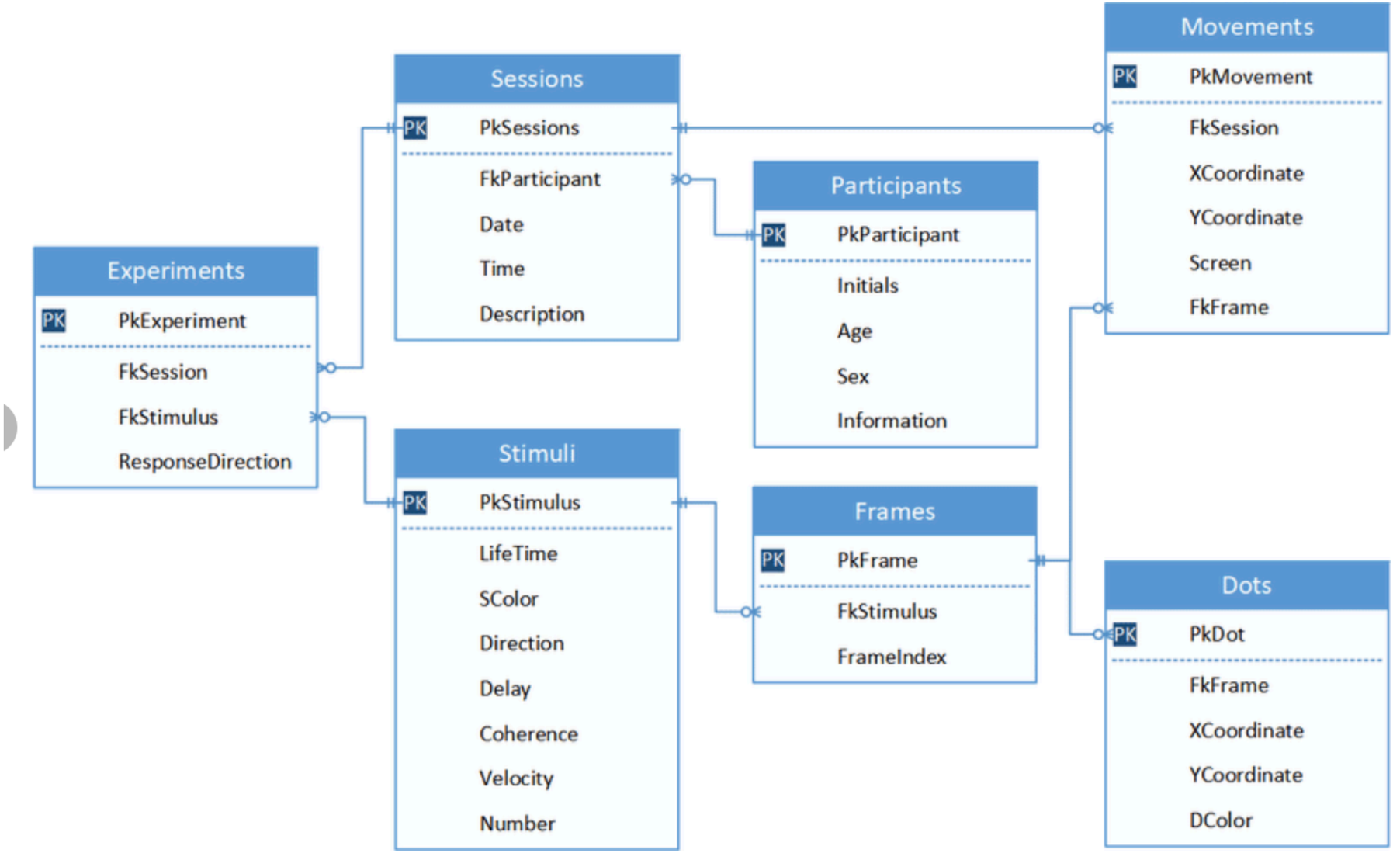


What is a Relational Database?

A relational database is a type of database that stores and provides access to data points that are related to one another. Relational databases are based on the relational model, an intuitive, straightforward way of representing data in tables. In a relational database, each row in the table is a record with a unique ID called the key. The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points.

Source: https://www.oracle.com/database/what-is-a-relational-database/

Structure of a relational DB



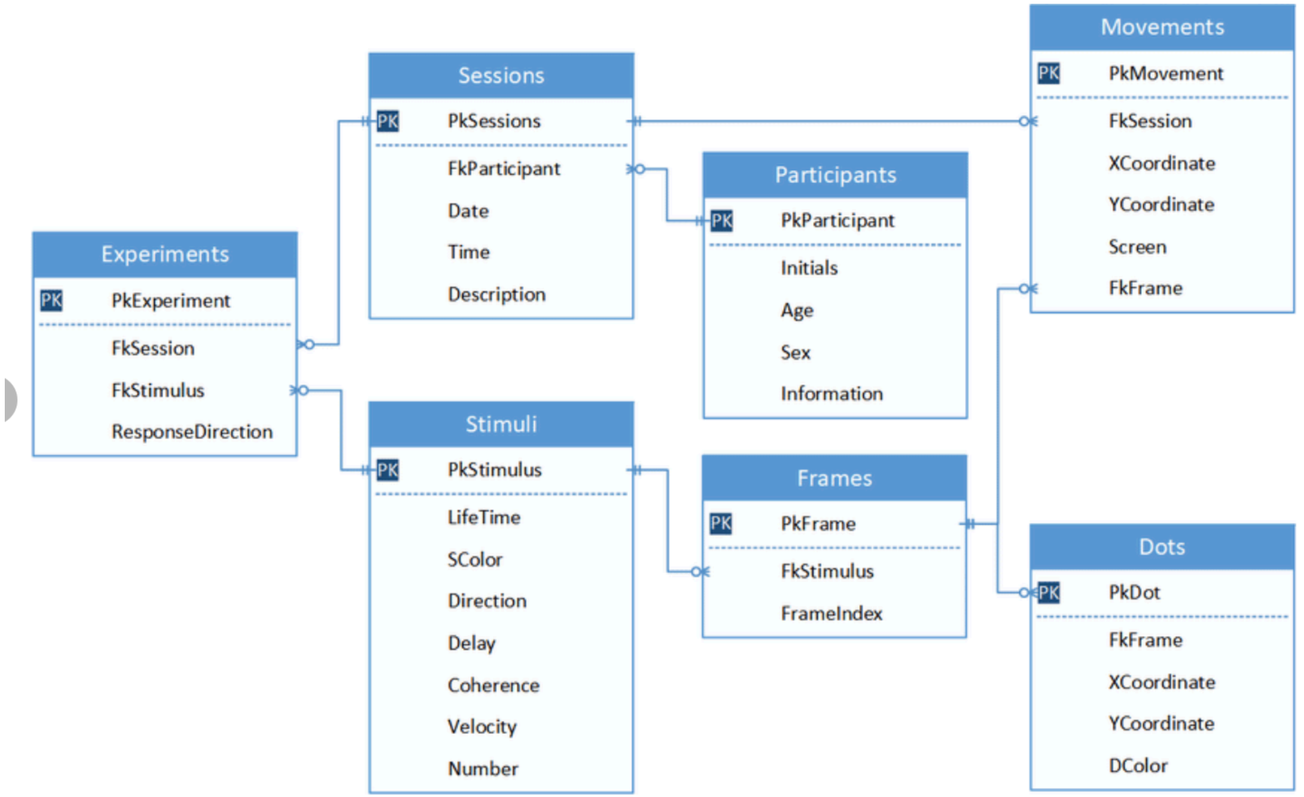
Basic principle:

NO Duplication

Why Not?

Structure of RDB

- A set of tables
  - A table consists of a set of rows where each row represents a unique entity
  - each table has a fixed set of columns which describe the attributes of the entities.
  - each table has zero or more **primary keys**
    - primary keys must be unique between entities (rows)
  - tables are typically linked so that a primary key of one table is a **foreign key** in other tables
    - foreign keys must have values in the primary key



Registrar DB

Student

ID

First Name

Last Name

...

How do you put students in classes?

Class Instance

Semester

Section

Instructor

Time

Place

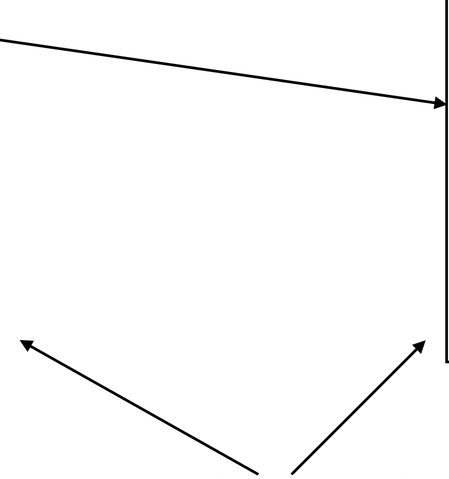
Class Listing

Department

Number

Description

Pre-reqs (??!)



What is the primary key?

This is incomplete as possibly wrongf, so complete it.

Mysql at Bryn Mawr

- **Available only on comet**
  - ssh YOU@comet.cs.brynmawr.edu
  - all files mount as if on powerpuff...

- To start mysql
  - COMETUNIX> mysql

- To exit mysql
  - ‘exit’ or CTRL-D

- batch mysql
  - make a file of mysql commands (typically xxx.sql)
  - COMETUNIX> mysql < xxx.mysql

Getting data from a RDBMS

- SQL!
  - Convenient to think of this a “Structured Query Language”
- For the remainder of this class, we will assume you are sitting at console and just want to get at information

- UNIX> mysql
  - sign in with your UNIX id (mysql -u DBUSERNAME)
  - actually starts MariaDB; fully open-source mySQL
  - `show databases;`
    - ; all SQL commands end with it
    - This lists all DBs which you can read.
  - `use flight;`
    - select this database to be the one you get data from
    - A DBMS probably has lots of databases (most of which you cannot see)
    - shortcut: UNIX> mysql -D test
  - `show tables;`
    - list the tables in the database
  - `describe aTable;`
    - show the columns of a table
  - `exit`
    - No semicolon needed!

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| test               |
+--------------------+
2 rows in set (0.002 sec)
```

```
MariaDB [flight]> show tables;
+------------------+
| Tables_in_flight |
+------------------+
| airports         |
| carriers         |
| flights          |
+------------------+
3 rows in set (0.000 sec)
```

```
MariaDB [test]> describe airports;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Name    | varchar(60) | NO   | PRI |         |       |
| Country | varchar(50) | YES  |     | NULL    |       |
| TLA     | char(3)     | NO   | PRI |         |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.001 sec)
```

Complete Select

```
SELECT selection_list          # What columns to select
FROM table_list                # Where to select rows from
WHERE primary_constraint       # What conditions rows must satisfy
GROUP BY grouping_columns      # How to group results
ORDER BY sorting_columns       # How to sort results
HAVING secondary_constraint    # Second condition rows must satisfy
LIMIT count;                   # Limit on results
```

More SQL

using select statements

- Simplest SQL is of form “select colA, colB,... from table”;
  - select name from carriers;
  - select * from carriers; // * means all columns
  - Column aliases
    - select name as nombre from carriers;
      - aliases create convenient names for use later in query and change column headers in output
- Restricting with where
  - select * from flights;
  - select * from flights where carrier=‘DL’;
    - default case insensitive

```
MariaDB [flight]> describe airports;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Name    | varchar(60) | YES  |     |         |       |
| Country | varchar(50) | YES  |     | NULL    |       |
| TLA     | char(3)     | NO   | PRI |         |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.001 sec)
```

```
MariaDB [flight]> describe carriers;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| Acronym  | varchar(3)  | NO   | PRI |         |       |
| Name     | varchar(50) | YES  |     | NULL    |       |
| CallSign | varchar(50) | YES  |     | NULL    |       |
| Country  | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.001 sec)
```

```
MariaDB [flight]> describe flights;
+----------------+------------+------+-----+-----------+-------+
| Field          | Type       | Null | Key | Default   | Extra |
+----------------+------------+------+-----+-----------+-------+
| Date           | date       | YES  |     | curdate() |       |
| DepartureTime  | time       | YES  |     | curtime() |       |
| ArrivalTime    | time       | YES  |     | curtime() |       |
| Carrier        | varchar(3) | YES  | MUL | NULL      |       |
| FlightNum      | int(11)    | YES  |     | -1        |       |
| ArrivalDelay   | int(11)    | YES  |     | 0         |       |
| DepartureDelay | int(11)    | YES  |     | 0         |       |
| Origin         | char(3)    | YES  | MUL | NULL      |       |
| Destination    | char(3)    | YES  | MUL | NULL      |       |
| Distance       | int(11)    | YES  |     | 0         |       |
| Cancelled      | tinyint(1) | YES  |     | 0         |       |
+----------------+------------+------+-----+-----------+-------+
11 rows in set (0.001 sec)
```

comparisons

- Boolean operators: and, or
- like
  - %==any number of characters
  - _ == one character
  - `select * from flights where Destination like 'E%';`
- rlike
  - egrep==rlike, fgrep==like
  - NOT standard SQL, but in SQL for MariaDB
  - more or less POSIX regular expressions
  - `select * from flights where Destination rlike 'e[xyz]r';`
- in — a set of possibilities
  - Could be done with OR
  - also `not in`
  - `select * from flights where Destination in ('ewr', 'ord');`

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal. **Note:** In some versions of SQL this operator may be written as !=
BETWEEN	Between a certain range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

Continuing SQL Select

- Date/time manipulation:
  - `select * from flights where Month(Date)=1 and day(date)=2 and hour(departuretime)=14;`
- Getting counts — using count
  - `select count(*) from flights where Month(Date)=1 and day(date)=2 and hour(departuretime)=14;`
- Getting distinct items `Select Distinct`
  - if specification includes primary key then select distinct will return everything
    - primary key(s) are defined to be distinct.
  - `select distinct origin from flights where Month(Date)=1 and day(date)=2 and hour(departuretime)=14;`
  - `select count(distinct origin) from flights where Month(Date)=1 and day(date)=2 and hour(departuretime)=14;`

Functions

an incomplete list

String Functions

ASCII	LEN	RTRIM
CHAR	LOWER	SPACE
CHARINDEX	LTRIM	STR
CONCAT	NCHAR	STUFF
Concat with +	PATINDEX	SUBSTRING
DATALENGTH	REPLACE	UPPER
LEFT	RIGHT	

Numeric/Math Functions

ABS	FLOOR	ROUND
AVG	MAX	SIGN
CEILING	MIN	SUM
COUNT	RAND	

Date/Time Functions

CURRENT_TIMESTAMP	DATEPART	MONTH
DATEADD	DAY	YEAR
DATEDIFF	GETDATE	
DATENAME	GETUTCDATE	

Grouping, ordering and limits

- Consider questions like:
  - How many flights came from each origin?
  - How many times does each flight number appear?
    - what are the 10 most frequently used flight numbers?
- The solution to these questions is the “group by” part of the select statement.
  - select count(*), origin from flights group by origin;
  - select count(*), flightnum from flights group by flightnum;
    - select count(*) as cou, flightnum from flights group by flightnum order by cou desc limit 10;

Exercises

Instructions: If you cannot think of a query for the whole answer, break down the question and write a query for some portion of the answer. Typically this will have more data than you need.

- List all departure delays
- List all unique departure delays
- What was the largest departure delay?
  - and what flight number(s) had that delay?
- What is the smallest arrival delay?
  - actual delay, not early or on-time
  - How main times did it happen?
  - On which flight numbers did this arrival delay happen?
- Which is greater, the number of destinations or the number of origins?
  - Use two queries
- How many carriers are in this database. How many actually fly into PHL? (Out of PHL?)
- How many carriers fly from each location from which you can fly directly to PHL?

```
select count(distinct carrier),origin from flights group by origin;
```

- How many flights go past midnight?

```
select  count(*) from flights where hour(arrivaltime)>hour(departuretime);
```

- How many flights from each origin landed in PHL on the day after they took off?

```
select count(origin), origin from flights where origin!='phl' and hour(departuretime)>hour(arrivaltime) group by origin;
```

```
MariaDB [flight]> describe airports;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Name    | varchar(60) | YES  |     |         |       |
| Country | varchar(50) | YES  |     | NULL    |       |
| TLA     | char(3)     | NO   | PRI |         |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.001 sec)
```

```
MariaDB [flight]> describe carriers;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| Acronym  | varchar(3)  | NO   | PRI |         |       |
| Name     | varchar(50) | YES  |     | NULL    |       |
| CallSign | varchar(50) | YES  |     | NULL    |       |
| Country  | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.001 sec)
```

```
MariaDB [flight]> describe flights;
+----------------+------------+------+-----+-----------+-------+
| Field          | Type       | Null | Key | Default   | Extra |
+----------------+------------+------+-----+-----------+-------+
| Date           | date       | YES  |     | curdate() |       |
| DepartureTime  | time       | YES  |     | curtime() |       |
| ArrivalTime    | time       | YES  |     | curtime() |       |
| Carrier        | varchar(3) | YES  | MUL | NULL      |       |
| FlightNum      | int(11)    | YES  |     | -1        |       |
| ArrivalDelay   | int(11)    | YES  |     | 0         |       |
| DepartureDelay | int(11)    | YES  |     | 0         |       |
| Origin         | char(3)    | YES  | MUL | NULL      |       |
| Destination    | char(3)    | YES  | MUL | NULL      |       |
| Distance       | int(11)    | YES  |     | 0         |       |
| Cancelled      | tinyint(1) | YES  |     | 0         |       |
+----------------+------------+------+-----+-----------+-------+
11 rows in set (0.001 sec)
```