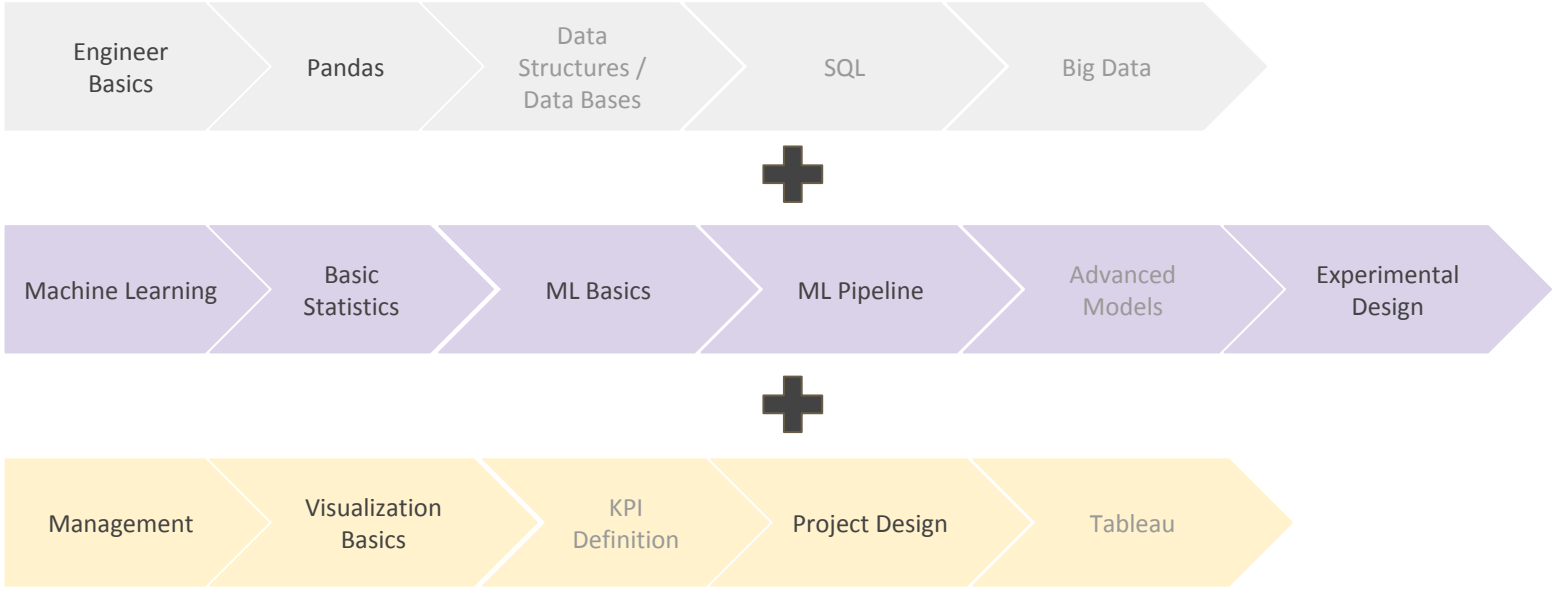
SQL for Data Scientists

Intro

Course Overview



Why should I care about SQL?

"The role of a Data Scientist is to turn raw data into actionable insights"

- ➔ Data exploration
- ➔ Data manipulation
- ➔ Collect and prepare data for analysis
- ➔ It's one of the most required skills for Data Scientists in 2017 (according to forbes.com)

interesting, but.. is SQL useful for my daily work?

SQL can be used to query a wide range of data volumes (from SQLite to BigQuery)

Data Scientist/Engineer tools supporting SQL:

- R
- RDBMSs: PostgreSQL, MySQL, MSSQL, Oracle
- Cassandra
- HiveSQL (Hadoop)
- BigQuery
- Spark SQL
- Presto

What we are NOT covering today

- How to install a SQL database
- Data modification
- Importing data into a database

What is SQL?

SQL (S-Q-L or "sequel") stands for *Structured Query Language*

Is a language used in programming and **designed for managing data** held in a relational database management system (RDBMS)

SQL is an ANSI standard, but there are dialects:

- Standard SQL
- PostgreSQL (PL/pgSQL)
- MySQL (SQL/PSM)
- Microsoft SQLServer (MSSQL)
- Oracle (PL/SQL)

What changes from dialect to dialect? Most commonly the data types and the functions

What can I do with SQL?

- Query data ← we are going to focus on this one

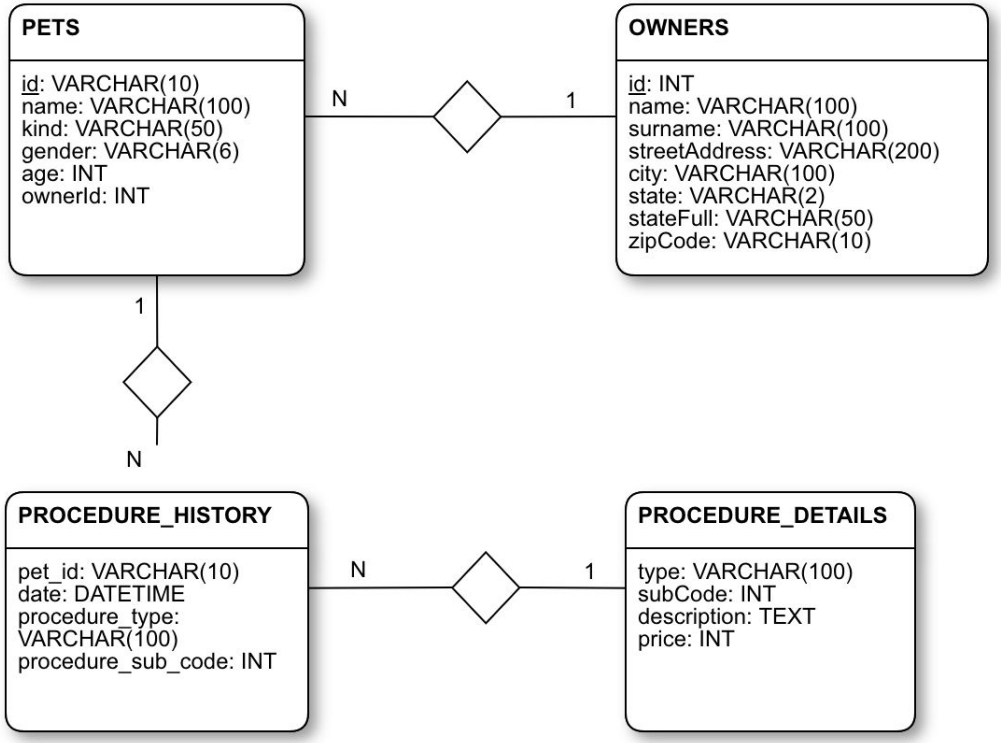
- Modify data:
  - Insert records
  - Update records
  - Delete records

- Modify Schema:
  - Insert/Update/Delete tables
  - Insert/Update/Delete fields

SQL Syntax: SELECT

SELECT *column1, column2, ..*
FROM *table*
WHERE *predicate(s)*
GROUP BY *column1, column2*
HAVING *predicate(s)*
ORDER BY *column1 [ASC/DESC], column2 [ASC/DESC]*
LIMIT 10

Example: Data Model



Data Types

- Numeric (INTEGER, BIGINT, DOUBLE PRECISION)
- Character (VARCHAR, TEXT)
- Date/Time (DATE, TIME, TIMESTAMP)
- Boolean (boolean)

Further info about PostgreSQL Data Types

Example: First try

- Open the Jupyter notebook in
- Let's try some examples:

Get all distinct pet kinds:	SELECT DISTINCT kind FROM pets;
Get all distinct procedure types:	SELECT DISTINCT `type FROM procedure_details;
Count pets:	SELECT COUNT(*) FROM pets;
Get the 5 oldest pets:	SELECT * FROM pets ORDER BY age ASC LIMIT 5;
Get the 1 most expensive procedure:	SELECT * FROM procedure_details ORDER BY price DESC LIMIT 1;

SQL Syntax: Filtering Results

SELECT *
FROM pets
WHERE *predicate(s)*

For example:

(Get all dogs)

```
SELECT id, name, age
FROM pets
WHERE kind = “Dog”;
```

(Get all dogs older than 5 years)

```
SELECT id, name, age
FROM pets
WHERE kind = “Dog” AND age > 5;
```

(Get all pets which age is between 3 and 5)

```
SELECT id, name, age
FROM pets
WHERE age BETWEEN 3 AND 5;
```

(Get all dogs or cats older than 5 years)

```
SELECT id, name, age
FROM pets
WHERE kind IN (“Dog”, “Cat”) AND age > 5;
```

SQL Syntax: Basic Operators

Operator	Description	Example
`=`	Equal to	`Author = 'Alcott'`
`<>`	Not equal to (many DBMSs accept `!=` in addition to `<>`)	`Dept <> 'Sales'`
`>`	Greater than	`Hire_Date > '2012-01-31'`
`<`	Less than	`Bonus < 50000.00`
`>=`	Greater than or equal	`Dependents >= 2`
`<=`	Less than or equal	`Rate <= 0.05`
`BETWEEN`	Between an inclusive range	`Cost BETWEEN 100.00 AND 500.00`
`LIKE`	Match a character pattern	`First_Name LIKE 'Will%'`
`IN`	Equal to one of multiple possible values	`DeptCode IN (101, 103, 209)`
`IS [NOT] NULL`	Compare to null (missing data)	`Address IS NOT NULL`
`IS [NOT] TRUE` *or* `IS [NOT] FALSE`	Boolean truth value test	`PaidVacation IS TRUE`
`IS NOT DISTINCT FROM`	Is equal to value or both are nulls (missing data)	`Debt IS NOT DISTINCT FROM - Receivables`
`AS`	Used to change a column name when viewing results	`SELECT employee AS "department1"`

Example: Filter out outliers

Filter out pets that are not Cats or Dogs:

```
SELECT *
FROM pets
WHERE kind NOT IN(“Dog”, “Cat”);
```

Filter out Pets not having an age:

```
SELECT *
FROM pets
WHERE age IS NOT NULL;
```

Aggregate Functions

General Purpose:

- COUNT
- MAX
- MIN
- AVG
- SUM

For Statistics:

- covar_pop
- regr_slope
- regr_r2
- stddev
- variance

For Ordered-Sets:

- percentile_cont
- percentile_disc

Further info about SQL ANSI Aggregate Functions: SQL in a Nutshell, 3rd Edition - O'Reilly
Classification of SQL Aggregate Functions: PostgreSQL Aggregate Functions

Example: General Purpose Aggregations

What is the average age of dogs in the pet store?

```
SELECT AVG(age)
FROM pets
WHERE kind = “Dog”
```

Which is the most expensive Procedure and how much does it cost?

How many VACCINATIONS were done by the pet store in total?

What is the average VACCINATIONS Procedure cost?

The LIKE Operator

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards used in conjunction with the LIKE operator:

% - The percent sign represents zero, one, or multiple characters
_ - The underscore represents a single character

Example: Get all Pet Owners which surname starts with H

```
SELECT * FROM owners WHERE surname LIKE 'H%'
```

Example: Get all Pets whose name has an “a” in the second character

```
SELECT * FROM pets WHERE name LIKE '_a%'
```

Example: LIKE

- How many dogs have a name starting with ‘B’ or ‘C’?
- How many dogs have a name that doesn’t start with ‘B’?
- Which pet owners have both name and surname starting with letter ‘C’?

Date Functions

Data Types:

- DATE (e.g. ‘2018-04-04’)
- TIME (e.g. ‘19:00:00’)
- DATETIME (e.g. ‘2018-04-04 19:00:00’)
- TIMESTAMP (e.g. ‘2018-04-04 19:00:00’)

Get current time:

current_date, current_time, current_timestamp

Extract parts of the date

EXTRACT(YEAR FROM `date`)
EXTRACT(MONTH FROM `date`)
EXTRACT(DAY FROM `date`)
EXTRACT(HOUR FROM `date`)

(in SQLite)
strftime(“%Y”, date)
strftime(“%m”, date)
strftime(“%d”, date)
strftime(“%H”, date)

More info about strftime in SQLite here

Example: Date Functions

- Get the days in the month where VACCINATIONS are done for dogs
- Get all Procedures types done after the day 15 of the month

Grouping Results

SELECT column, aggregated_function
FROM table
GROUP BY groups rows sharing a property so that an aggregate function can be applied to each group
HAVING selects among the groups defined by the GROUP BY clause

Example: Count the number of pets per kind:

```
SELECT kind, COUNT(*)
FROM pets
GROUP BY kind
```

Example: Count the number of pets per kind, having more than 10 pets:

```
SELECT kind, COUNT(*)
FROM pets
GROUP BY kind
HAVING COUNT(*) > 15
```

Example: Aggregate Functions with Grouping

- Get the average price, max and min for each Procedure type
- Get the number of Procedures per month
- Get the number of Procedures of each type per month

Joins

A SQL Join clause is a way to retrieve information from two or more tables in a database.

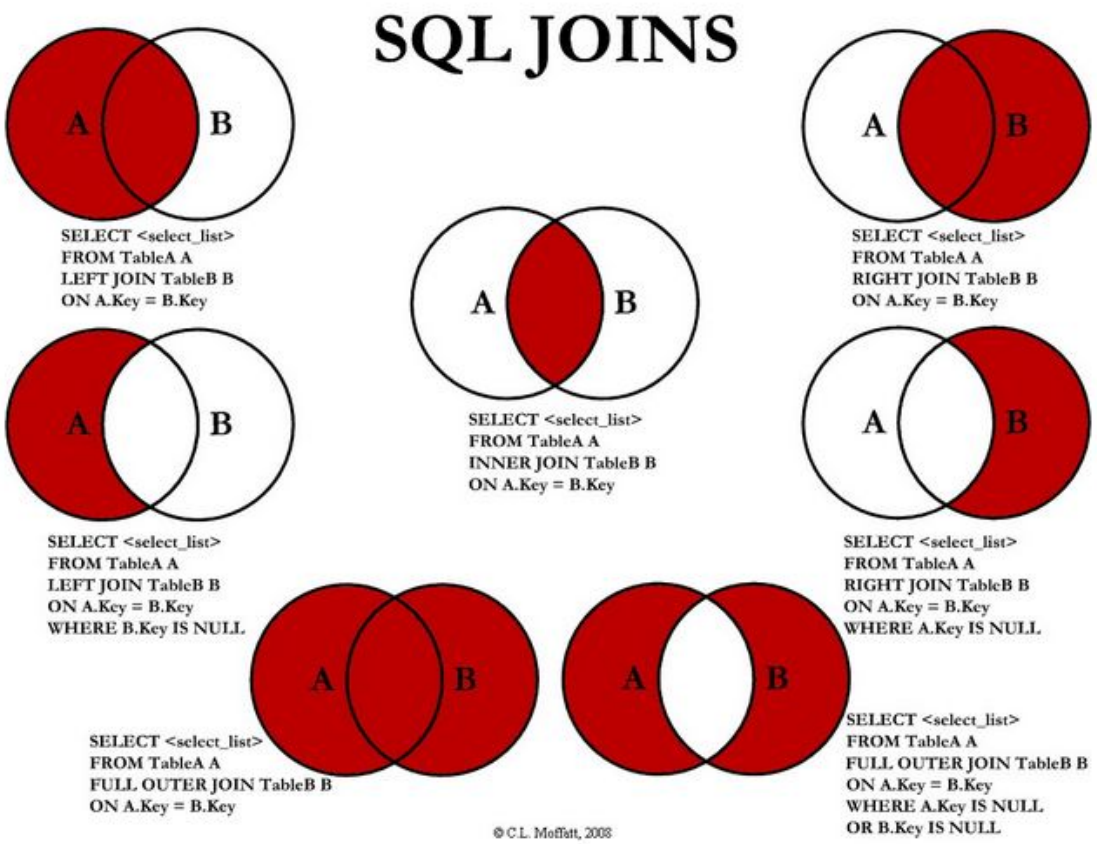
How does it work?

Example: Get the name of dog owners

```
SELECT o.name AS owner_name
FROM pets p
INNER JOIN
owners o ON (p.owner_id = o.id)
WHERE p.kind = “Dog”
ORDER BY o.name ASC
```

Type of Joins

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL OUTER JOIN



Examples: JOINs

- Get pets that had Procedures
- Get owners of pets with Procedures
- Get pets without Procedures
- Get the amount of money coming from VACCINATIONS (by month)
- Get owners having more than 1 pet
- Check which Procedures are not in the history

Unions

UNION combines the results of two SQL queries into a single table of all matching rows.

It comes in 2 flavours:

- Results are deduplicated (UNION)
- Results are deduplicated (UNION ALL)

Example: UNION

- Get all names from owners or pets:

```
SELECT name AS subject_name
FROM pets
ORDER BY name ASC

UNION (ALL)

SELECT name AS subject_name
FROM owners
ORDER BY name ASC;
```

Exercises

- What kind of pet have more procedures overall?
- Can we infer the pet kind from the owner location?
- Can we infer the pet age from the Procedure history?
- How much revenue per month is brought by each procedure?
- Which pet kind and gender bring more revenue?
- Is there any seasonality on the Procedures?

Recapping..

- ➔ What we learned today
- ➔ SQL for Data Scientist
- ➔ SQL for the course project

I like it! Where can I go next?

- w3schools SQL Tutorial: https://www.w3schools.com/sql/
- Practice online: https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all
- Intro to SQL for Data Science: https://www.datacamp.com/courses/intro-to-sql-for-data-science
- Advanced SQL for Data Scientists: https://www.lynda.com/SQL-tutorials/Advanced-SQL-Data-Scientists/559183-2.html