Intuition for SQL

Goals for

Today & Next week

(SQL, SQL, SQL)

Phase I: Intuition for SQL (1st half of today)

Basic Relational model (aka tables)

Example SQL (exploring real datasets)

Phase II: Formal description

SQL concepts we'll study (similar to Python map-reduce)

Schemas, Query structure of SELECT-FROM-WHERE, JOINs, etc

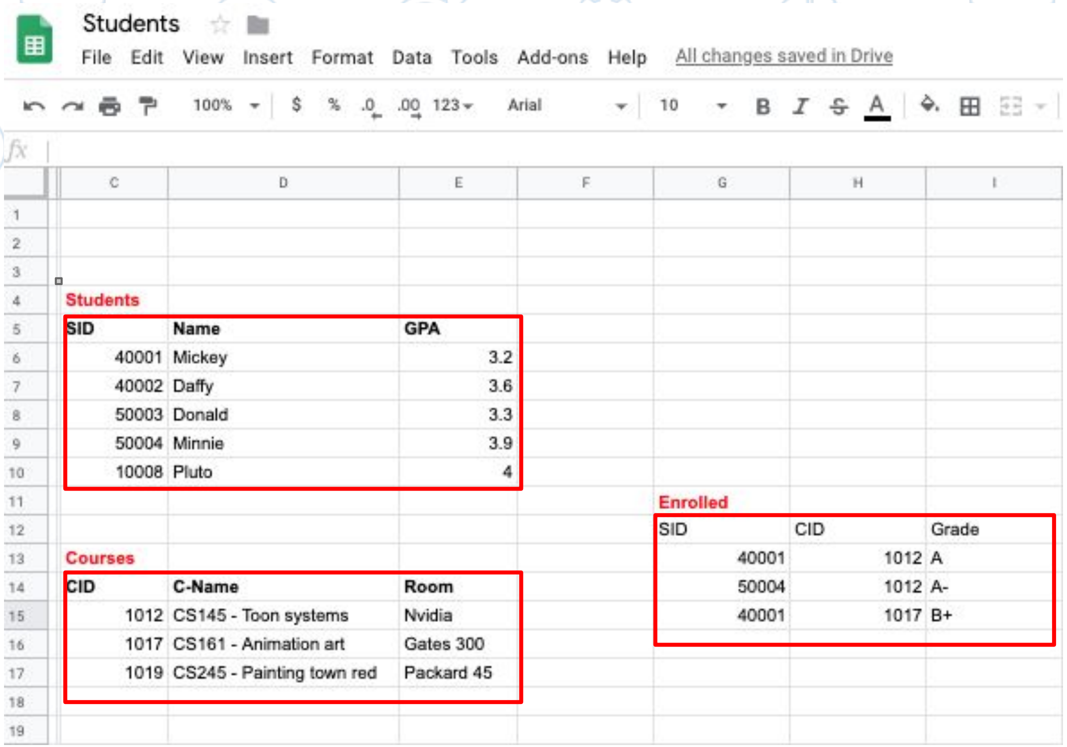
A Motivating Example

A basic Course Management System (CMS):

Entities (e.g., Students, Courses, Professors)

Relationships (e.g., *Who takes what, Who teaches what?*)

Simple DB == Spreadsheets



Students

SID	Name	GPA
40001	Mickey	3.2
40002	Daffy	3.6
50003	Donald	3.3
50004	Minnie	3.9
10008	Pluto	4

Courses

CID	C-Name	Room
1012	CS145 - Toon systems	Nvidia
1017	CS161 - Animation art	Gates 300
1019	CS245 - Painting town red	Packard 45

Enrolled

SID	CID	Grade
40001	1012	A
50004	1012	A-
40001	1017	B+

Tables

Student(sid: *string*, name: *string*, gpa: *float*)

Courses(cid: *string*, c-name: *string*, room: *string*)

Enrolled(sid: *string*, cid: *string*, grade: *string*)

sid: Connects Enrolled to Students

cid: Connects Enrolled to Courses

Queries [“compute” over tables]

- Minnie’s GPA?
- AVG student GPA?
- Mickey’s classes?
- AVG student GPA in CS145?

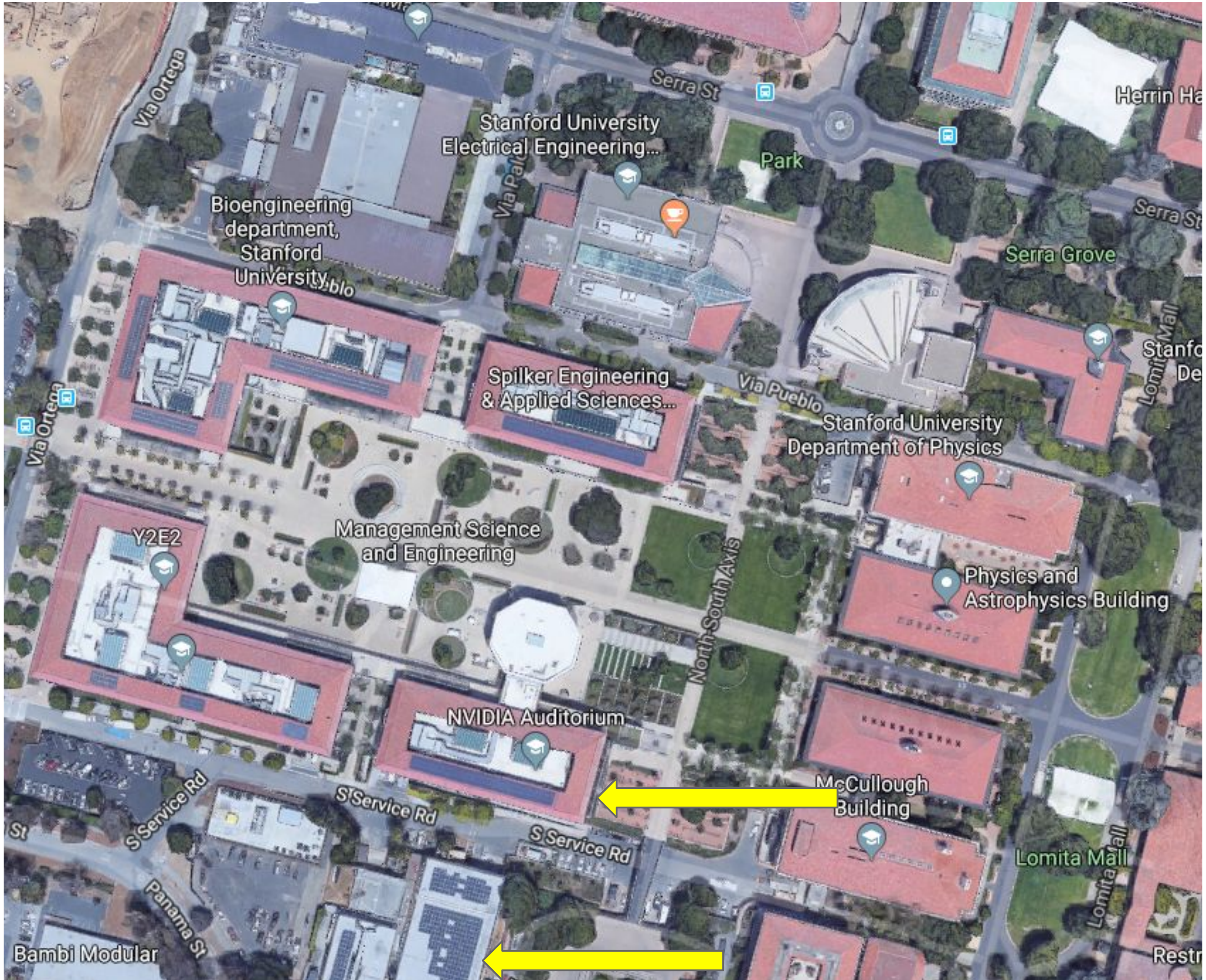
Example SQL

Part I

Example 1

Sun Roof potential

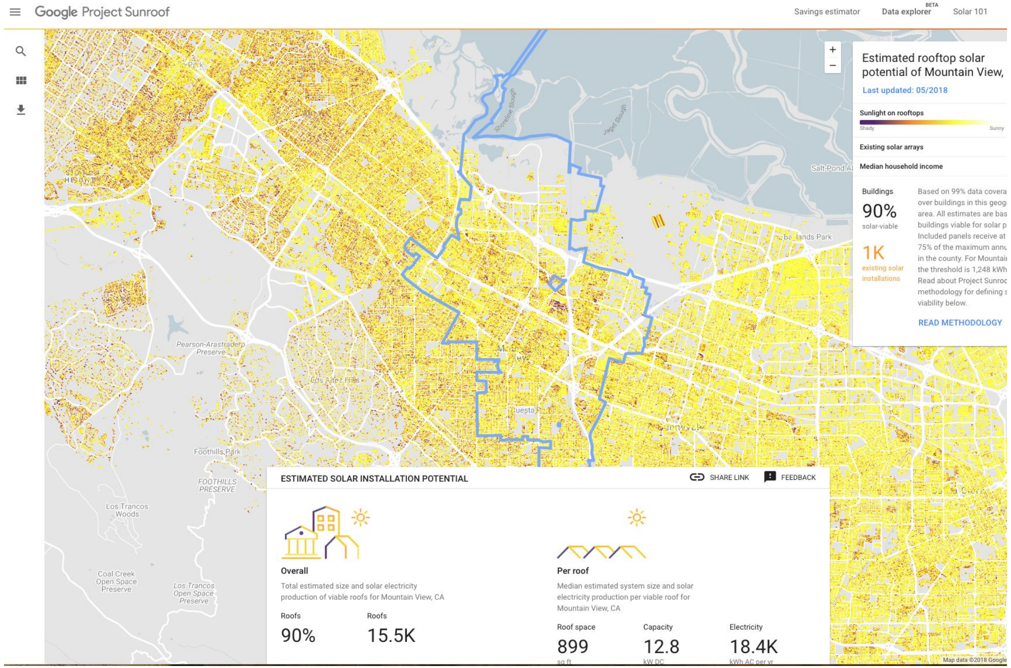
from Satellite images



Example 1

SunRoof potential

SunRoof explorer



Estimated rooftop solar potential of Mountain View,

Last updated: 05/2018

Sunlight on rooftops

Shady — Sunny

Existing solar arrays

Median household income

Buildings

90%

solar-viable

1K

existing solar installations

Based on 99% data covera over buildings in this geog area. All estimates are bas buildings viable for solar p Included panels receive at 75% of the maximum annu in the county. For Mountai the threshold is 1,248 kWh Read about Project Sunroo methodology for defining s viability below.

READ METHODOLOGY

ESTIMATED SOLAR INSTALLATION POTENTIAL

SHARE LINK

FEEDBACK

Overall

Total estimated size and solar electricity production of viable roofs for Mountain View, CA

Roofs	Roofs
90%	15.5K

Per roof

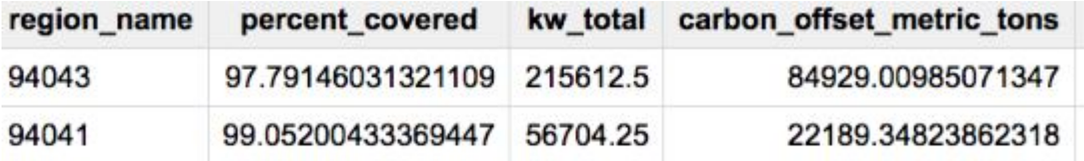
Median estimated system size and solar electricity production per viable roof for Mountain View, CA

Roof space	Capacity	Electricity
899 sq ft	12.8 kW DC	18.4K kWh AC per yr

Example 1

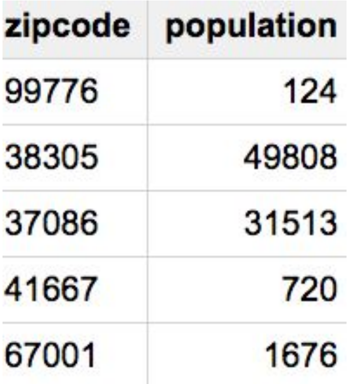
Public dataset

Public Dataset: Solar potential by postal code



region_name	percent_covered	kw_total	carbon_offset_metric_tons
94043	97.79146031321109	215612.5	84929.00985071347
94041	99.05200433369447	56704.25	22189.34823862318

Public Dataset: USA.population by zip2010



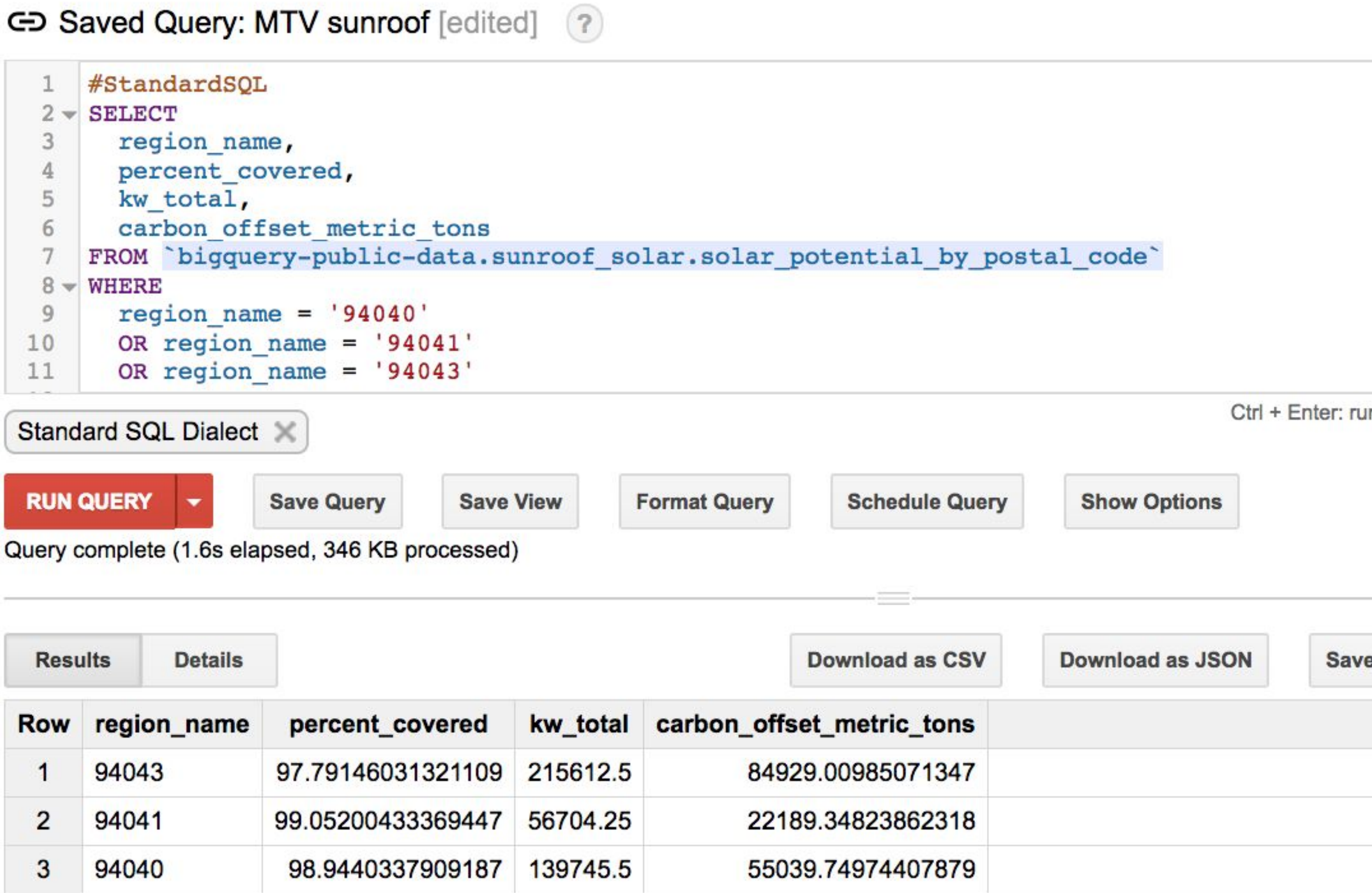
zipcode	population
99776	124
38305	49808
37086	31513
41667	720
67001	1676

Example 1

SunRoof

On BigQuery Public dataset

What is the solar potential of Mountain View, CA? [Run query]



Saved Query: MTV sunroof [edited]

```
#StandardSQL
SELECT
  region_name,
  percent_covered,
  kw_total,
  carbon_offset_metric_tons
FROM `bigquery-public-data.sunroof_solar.solar_potential_by_postal_code`
WHERE
  region_name = '94040'
  OR region_name = '94041'
  OR region_name = '94043'
```

Standard SQL Dialect

Query complete (1.6s elapsed, 346 KB processed)

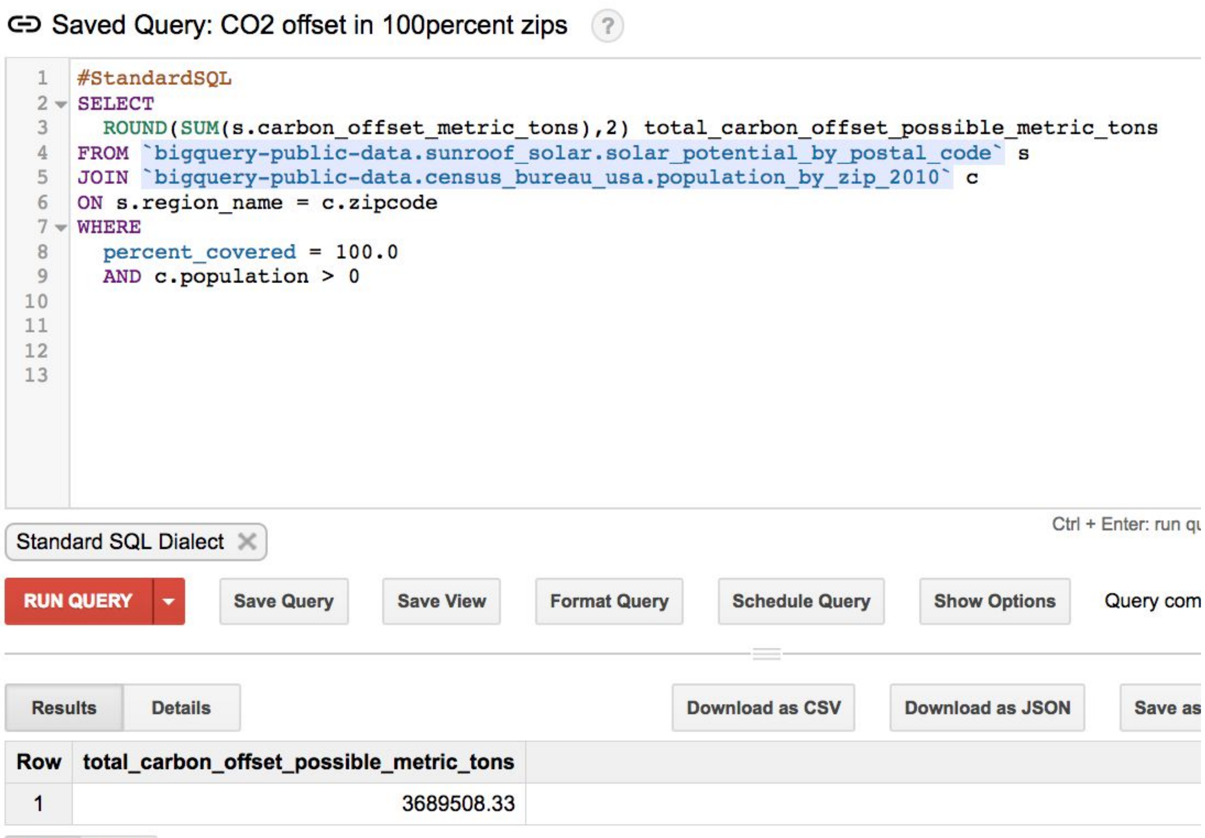
Row	region_name	percent_covered	kw_total	carbon_offset_metric_tons
1	94043	97.79146031321109	215612.5	84929.00985071347
2	94041	99.05200433369447	56704.25	22189.34823862318
3	94040	98.9440337909187	139745.5	55039.74974407879

Example 2

SunRoof

Public dataset
On BigQuery

How many metric tons of carbon would we offset, if building in communities with 100% coverage all had solar roofs? [Run query]



Saved Query: CO2 offset in 100percent zips

```
#StandardSQL
SELECT
  ROUND(SUM(s.carbon_offset_metric_tons),2) total_carbon_offset_possible_metric_tons
FROM `bigquery-public-data.sunroof_solar.solar_potential_by_postal_code` s
JOIN `bigquery-public-data.census_bureau_usa.population_by_zip_2010` c
ON s.region_name = c.zipcode
WHERE
  percent_covered = 100.0
  AND c.population > 0
```

Standard SQL Dialect ✕

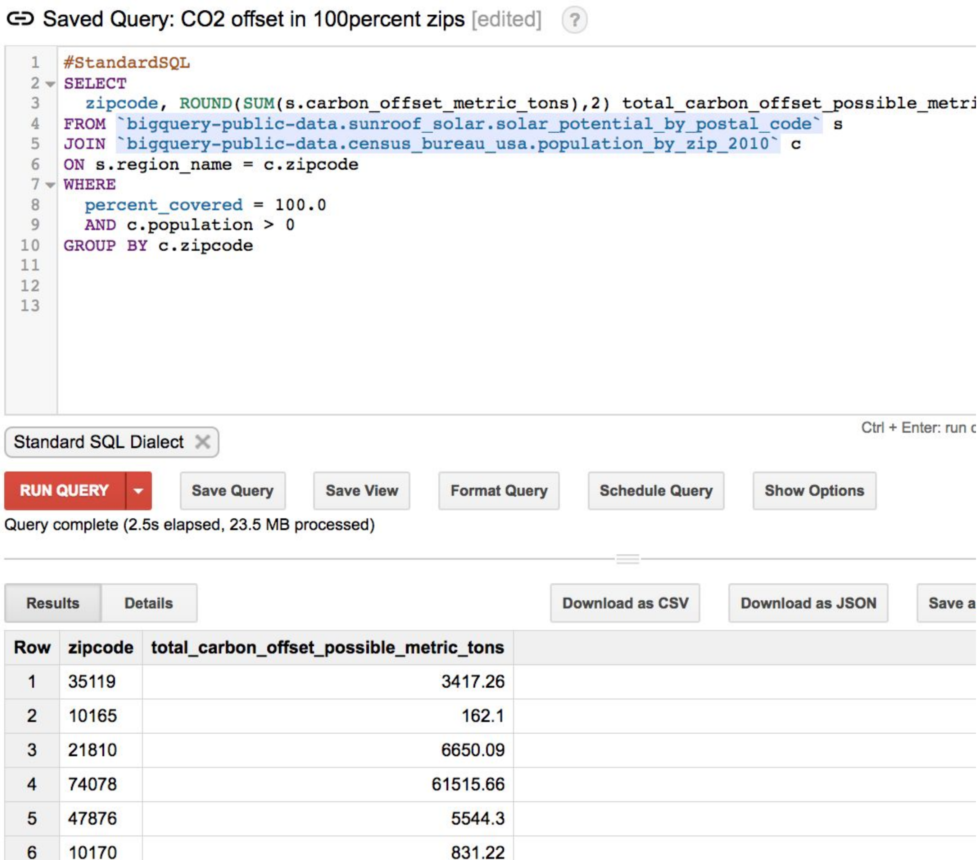
Row	total_carbon_offset_possible_metric_tons
1	3689508.33

Example 2

SunRoof

Public dataset On BigQuery

How many metric tons of carbon would we offset, per zipcode?



Saved Query: CO2 offset in 100percent zips [edited]

```
#StandardSQL
SELECT
  zipcode, ROUND(SUM(s.carbon_offset_metric_tons),2) total_carbon_offset_possible_metri
FROM `bigquery-public-data.sunroof_solar.solar_potential_by_postal_code` s
JOIN `bigquery-public-data.census_bureau_usa.population_by_zip_2010` c
ON s.region_name = c.zipcode
WHERE
  percent_covered = 100.0
  AND c.population > 0
GROUP BY c.zipcode
```

Standard SQL Dialect

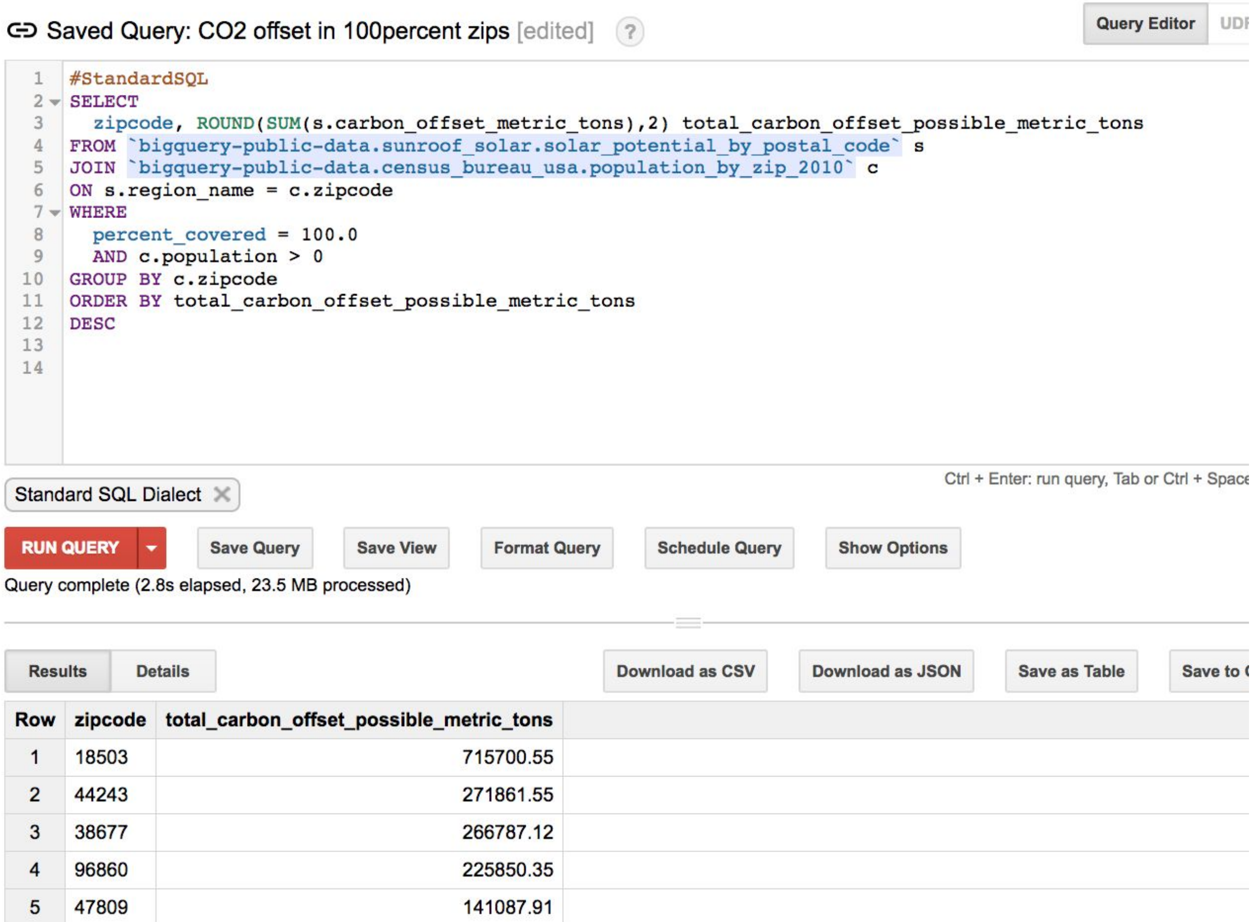
Query complete (2.5s elapsed, 23.5 MB processed)

Row	zipcode	total_carbon_offset_possible_metric_tons
1	35119	3417.26
2	10165	162.1
3	21810	6650.09
4	74078	61515.66
5	47876	5544.3
6	10170	831.22

Example 2

SunRoof

How many metric tons of carbon would we offset, per zipcode sorted?



```
#StandardSQL
SELECT
  zipcode, ROUND(SUM(s.carbon_offset_metric_tons),2) total_carbon_offset_possible_metric_tons
FROM `bigquery-public-data.sunroof_solar.solar_potential_by_postal_code` s
JOIN `bigquery-public-data.census_bureau_usa.population_by_zip_2010` c
ON s.region_name = c.zipcode
WHERE
  percent_covered = 100.0
  AND c.population > 0
GROUP BY c.zipcode
ORDER BY total_carbon_offset_possible_metric_tons
DESC
```

Standard SQL Dialect ✕

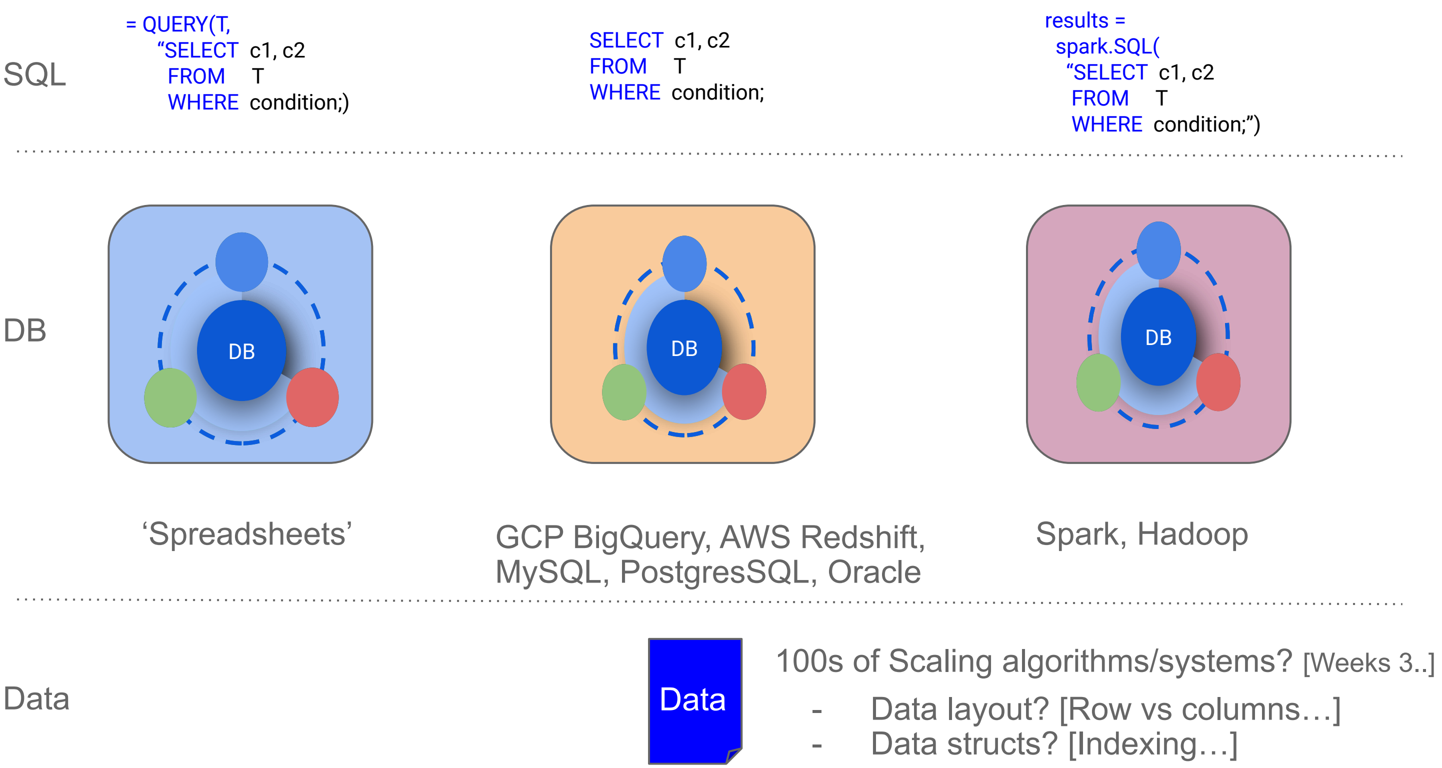
Query complete (2.8s elapsed, 23.5 MB processed)

Row	zipcode	total_carbon_offset_possible_metric_tons
1	18503	715700.55
2	44243	271861.55
3	38677	266787.12
4	96860	225850.35
5	47809	141087.91

Reminder

Special Databases

Query with SQL, universally over ‘all’ DBs



SQL

```
= QUERY(T,
    "SELECT  c1, c2
     FROM     T
     WHERE  condition;)
```

```
SELECT  c1, c2
FROM     T
WHERE  condition;
```

```
results =
  spark.SQL(
   "SELECT  c1, c2
    FROM     T
    WHERE  condition;")
```

DB

‘Spreadsheets’

GCP BigQuery, AWS Redshift, MySQL, PostgresSQL, Oracle

Spark, Hadoop

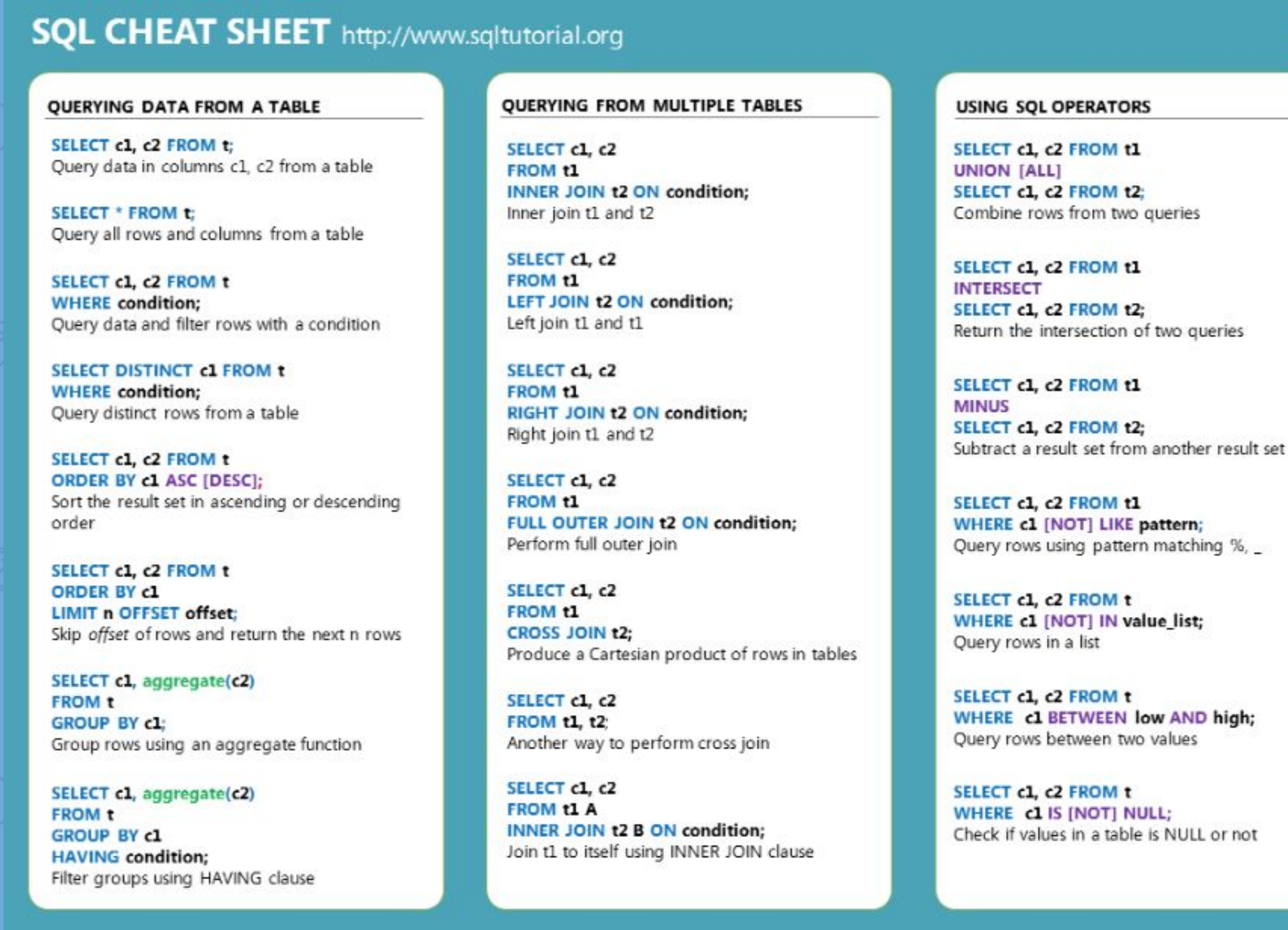
Data

100s of Scaling algorithms/systems? [Weeks 3..]

- Data layout? [Row vs columns…]
- Data structs? [Indexing…]

Preview

SQL queries



SQL CHEAT SHEET http://www.sqltutorial.org

QUERYING DATA FROM A TABLE

```
SELECT c1, c2 FROM t;
```
Query data in columns c1, c2 from a table

```
SELECT * FROM t;
```
Query all rows and columns from a table

```
SELECT c1, c2 FROM t
WHERE condition;
```
Query data and filter rows with a condition

```
SELECT DISTINCT c1 FROM t
WHERE condition;
```
Query distinct rows from a table

```
SELECT c1, c2 FROM t
ORDER BY c1 ASC [DESC];
```
Sort the result set in ascending or descending order

```
SELECT c1, c2 FROM t
ORDER BY c1
LIMIT n OFFSET offset;
```
Skip *offset* of rows and return the next n rows

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1;
```
Group rows using an aggregate function

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1
HAVING condition;
```
Filter groups using HAVING clause

QUERYING FROM MULTIPLE TABLES

```
SELECT c1, c2
FROM t1
INNER JOIN t2 ON condition;
```
Inner join t1 and t2

```
SELECT c1, c2
FROM t1
LEFT JOIN t2 ON condition;
```
Left join t1 and t1

```
SELECT c1, c2
FROM t1
RIGHT JOIN t2 ON condition;
```
Right join t1 and t2

```
SELECT c1, c2
FROM t1
FULL OUTER JOIN t2 ON condition;
```
Perform full outer join

```
SELECT c1, c2
FROM t1
CROSS JOIN t2;
```
Produce a Cartesian product of rows in tables

```
SELECT c1, c2
FROM t1, t2;
```
Another way to perform cross join

```
SELECT c1, c2
FROM t1 A
INNER JOIN t2 B ON condition;
```
Join t1 to itself using INNER JOIN clause

USING SQL OPERATORS

```
SELECT c1, c2 FROM t1
UNION [ALL]
SELECT c1, c2 FROM t2;
```
Combine rows from two queries

```
SELECT c1, c2 FROM t1
INTERSECT
SELECT c1, c2 FROM t2;
```
Return the intersection of two queries

```
SELECT c1, c2 FROM t1
MINUS
SELECT c1, c2 FROM t2;
```
Subtract a result set from another result set

```
SELECT c1, c2 FROM t1
WHERE c1 [NOT] LIKE pattern;
```
Query rows using pattern matching %, _

```
SELECT c1, c2 FROM t
WHERE c1 [NOT] IN value_list;
```
Query rows in a list

```
SELECT c1, c2 FROM t
WHERE  c1 BETWEEN low AND high;
```
Query rows between two values

```
SELECT c1, c2 FROM t
WHERE  c1 IS [NOT] NULL;
```
Check if values in a table is NULL or not

sqltutorial.org/sql-cheat-sheet