**set** search\_path **to** data\_sci;

**create table** data\_sci.org\_structure (

id integer,

department\_name text,

parent\_department\_id integer);

**insert into** data\_sci.org\_structure values

(1, 'CEO Office', null),

(2, 'VP Sales', 1),

(3, 'VP Operations', 1),

(4, 'Northeast Sales',2),

(5, 'Northwest Sales',2),

Joining

**select**

cd.department\_name, count(\*)

**from**

data\_sci.employees e

**join**

data\_sci.company\_departments cd

**on**

e.department\_id = cd.id

**group by**

cd.department\_name

**order by**

count(\*);

Subqueries

*Shows average salary by department*

**select**

e1.last\_name,

e1.salary,

e1.department\_id,

(**select avg**(salary) **from** data\_sci.employees e2 **where** e1.department\_id = e2.department\_id)

**from**

data\_sci.employees e1;

*Rows where salary greater than 100’000*

**select**

**round(avg**(e1.salary), 2)

**from**

(**select** \* from data\_sci.employees **where** salary > 100000) e1;

*Find all the department ids where at least one employee has the maximum salary*

**select**

department\_id

**from**

data\_sci.employees e1

**where**

(**select** max(salary) **from** data\_sci.employees e2) = e1.salary;

Filtering by subgroup with Having

**select**

cd.department\_name, count(\*)

**from**

data\_sci.employees e

**join**

data\_sci.company\_departments cd

**on**

e.department\_id = cd.id

**group by**

cd.department\_name

**having**

count(\*) > 50 *-- use having with aggregate values*

**order by**

cd.department\_name;

Grouping by totals and subtotals

*Count based by country name and country region.*

**select**

cr.country\_name, cr.region\_name, count(e.\*)

**from**

data\_sci.employees e

**join**

data\_sci.company\_regions cr

**on**

e.region\_id = cr.id

**group by**

**rollup**(cr.country\_name, cr.region\_name)

**order by**

cr.country\_name, cr.region\_name;

Partitioning (a form of grouping) with window functions

**select**

department\_id,

last\_name,

salary,

**first\_value**(salary) **over** (**partition by** department\_id **order by** salary **asc**) first\_sal

or - - **nth\_value**(salary,2) **over** (**partition by** department\_id **order by** salary **desc**)

**from**

data\_sci.employees;

**select**

department\_id,

last\_name,

salary,

**lead**(salary) **over** (**partition by** department\_id **order by** salary **desc**) - - or lag

**from**

data\_sci.employees;

*Look at the average salaries per department*

**select**

department\_id,

salary,

**avg**(salary) over (partition by department\_id)

**from**

data\_sci.employees;

**CTE**

**with** east\_regions **as**

(**select** id

**from** data\_sci.company\_regions cr

**where** region\_name like '%east%')

**select**

sum(salary),

round(avg(salary),2)

**from**

data\_sci.employees e2

**inner join** east\_regions

**on** east\_regions.id=e2.region\_id

**Formatting**

**select**

initcap(department\_name) - - capitalizing first letter

**from**

data\_sci.company\_departments;

**select**

ltrim(‘ kelly’) = ‘kelly’;

**select**

job\_title || last\_name -- concaternate job title and last name

OR,

**select**

concat(job\_title, '-',null)

**from**

data\_sci.employees;

**select** **distinct**

job\_titles

**from**

data\_sci.employees

**where**

job\_title **like** ‘vp%’ -- match any number of characters after vp;

**select distinct**

job\_titles

**from**

data\_sci.employees

**where**

job\_title **similar to** '(vp%|web%)' -- vp or web