Examples of how to extract and analyze data

crrt-notebook.ipynb (38 cells)

emergency-department-exploration.ipynb (7 cells)

first_labs.ipynb (6 cells)

tableone-demo.ipynb (6 cells)

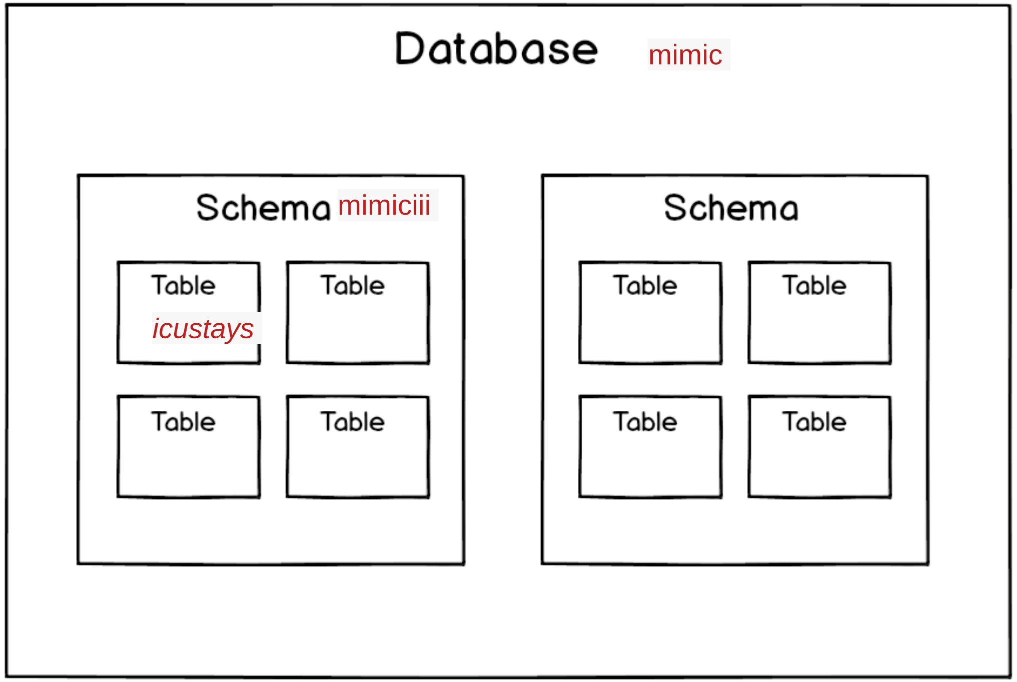
vancomycin-dosing.ipynb (11 cells)

ipynb_example/icu_length_of_stay.ipynb

rmd_consort_diag/plot_consort_diagram.Rmd

rmd_example/mimic_los_bigquery.Rmd. mimic_los_postgres.Rmd

+ tutorials/sql-intro.md



http://www.wagonhq.com/sql-tutorial/how-is-my-database-organized

first_labs.ipynb

SQL | WITH clause

https://modern-sql.com/feature/with Compatibility

https://www.geeksforgeeks.org/sql-with-clause/ Examples

https://github.com/MIT-LCP/mimic-code/blob/master/tutorials/sql-intro.md

The cursor class http://initd.org/psycopg/docs/cursor.html

```
cur.execute('SET search_path to {}'.format(schema_name))
```

SET https://www.postgresql.org/docs/9.1/sql-set.html

```
SET [ SESSION | LOCAL ] configuration_parameter { TO | = }
{ value | 'value' | DEFAULT }
```

http://www.sqltutorial.org/sql-window-functions/sql-rank/

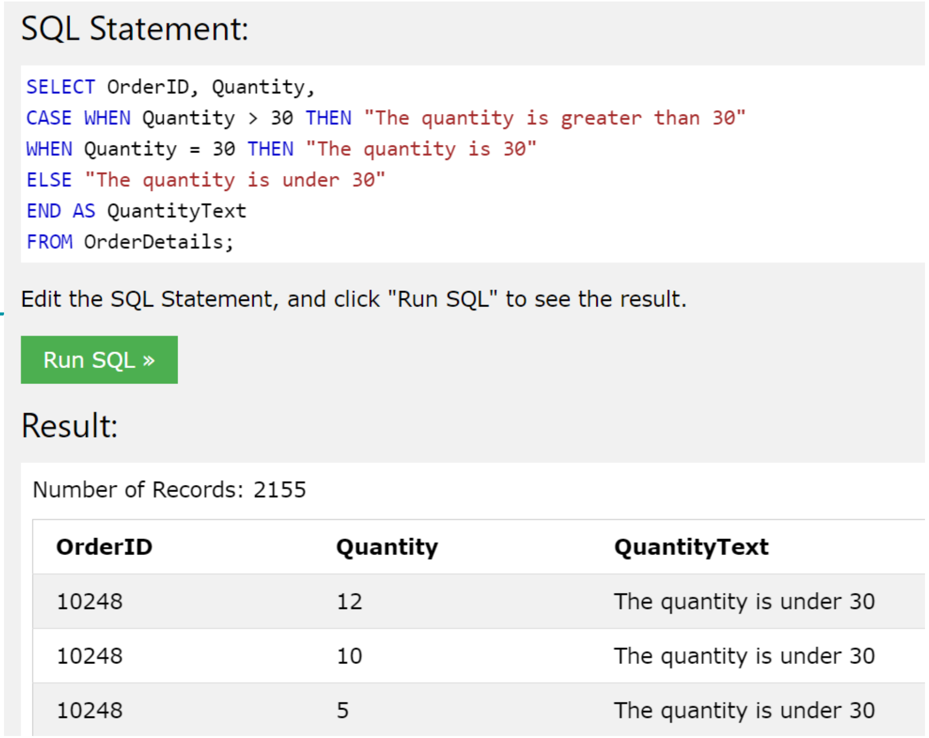
```
        RANK() OVER (
            PARTITION BY department_id
            ORDER BY salary) salary_rank
```

CASE statement for if/else logic

Every CASE statement must **end** with the **END** statement.

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

https://www.w3schools.com/sql/trymysql.asp?filename=trysql_



SQL Statement:

```
SELECT OrderID, Quantity,
CASE WHEN Quantity > 30 THEN "The quantity is greater than 30"
WHEN Quantity = 30 THEN "The quantity is 30"
ELSE "The quantity is under 30"
END AS QuantityText
FROM OrderDetails;
```

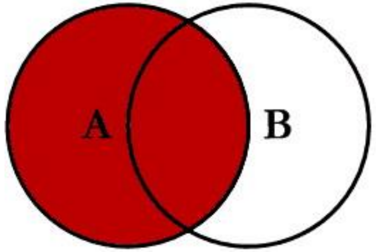
Edit the SQL Statement, and click "Run SQL" to see the result.

Result:

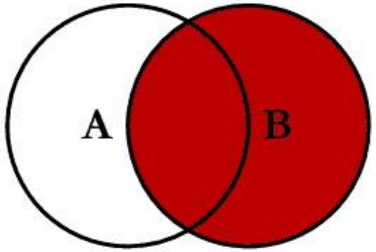
Number of Records: 2155

OrderID	Quantity	QuantityText
10248	12	The quantity is under 30
10248	10	The quantity is under 30
10248	5	The quantity is under 30

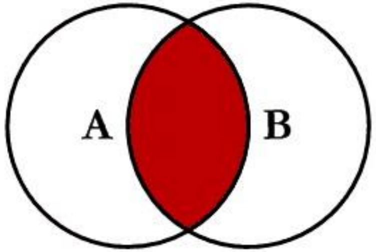
SQL JOINS



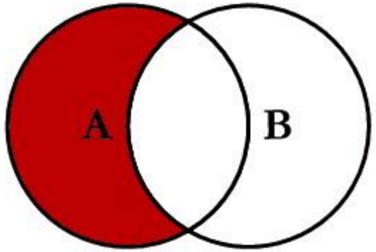
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



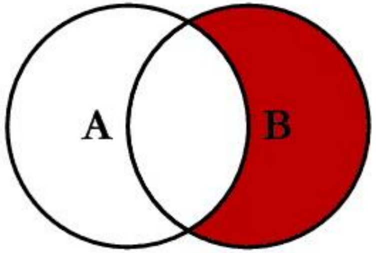
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



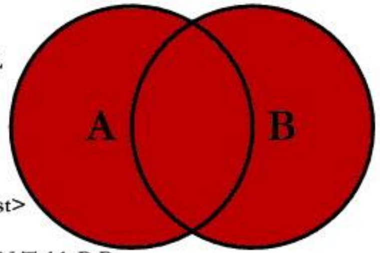
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



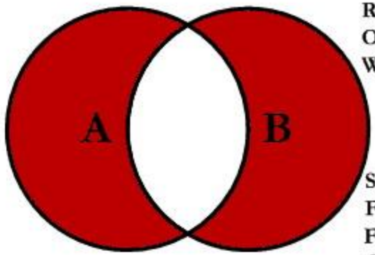
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

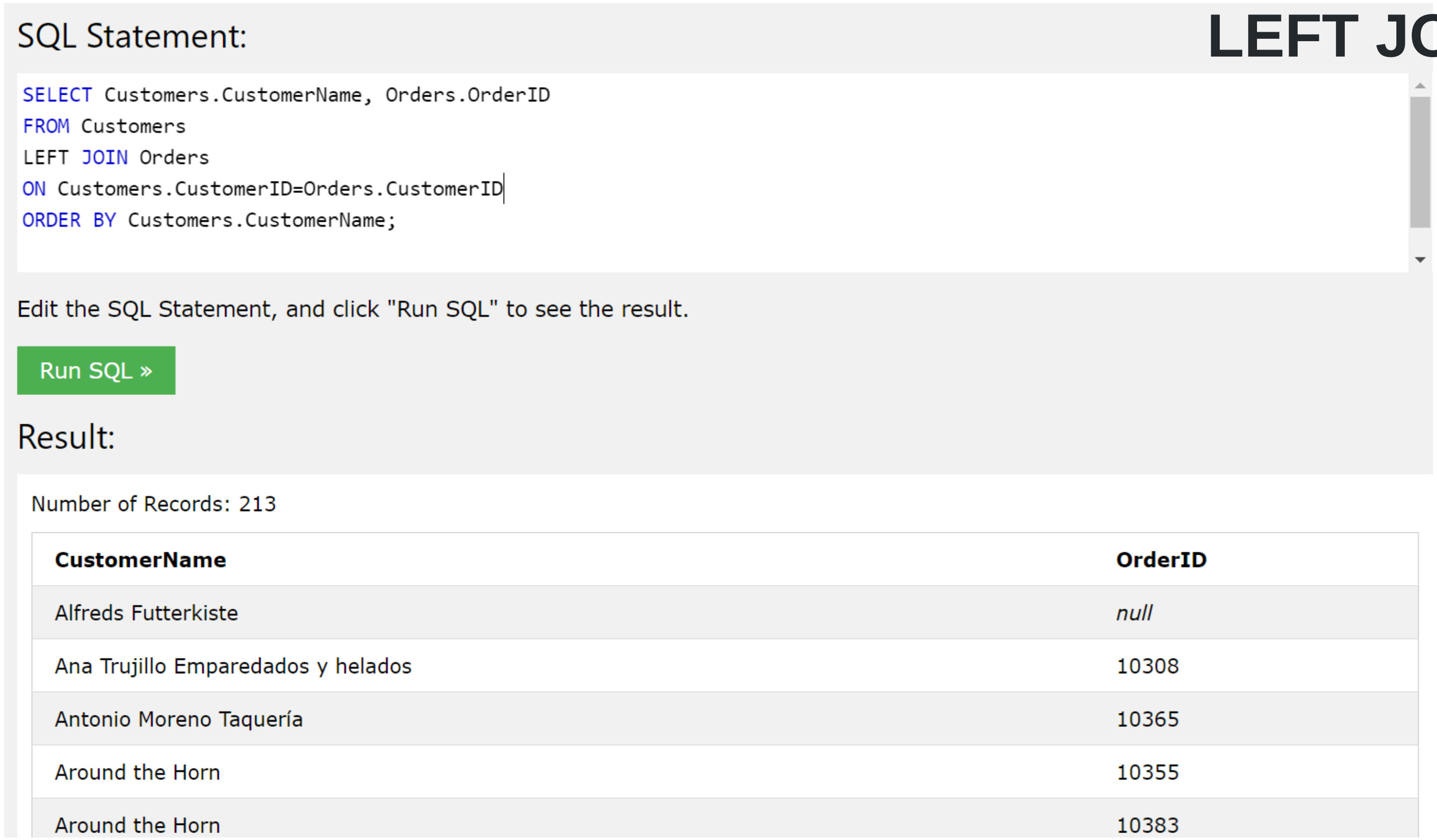


```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

LEFT JOIN



SQL Statement:

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
LEFT JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
ORDER BY Customers.CustomerName;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Result:

Number of Records: 213

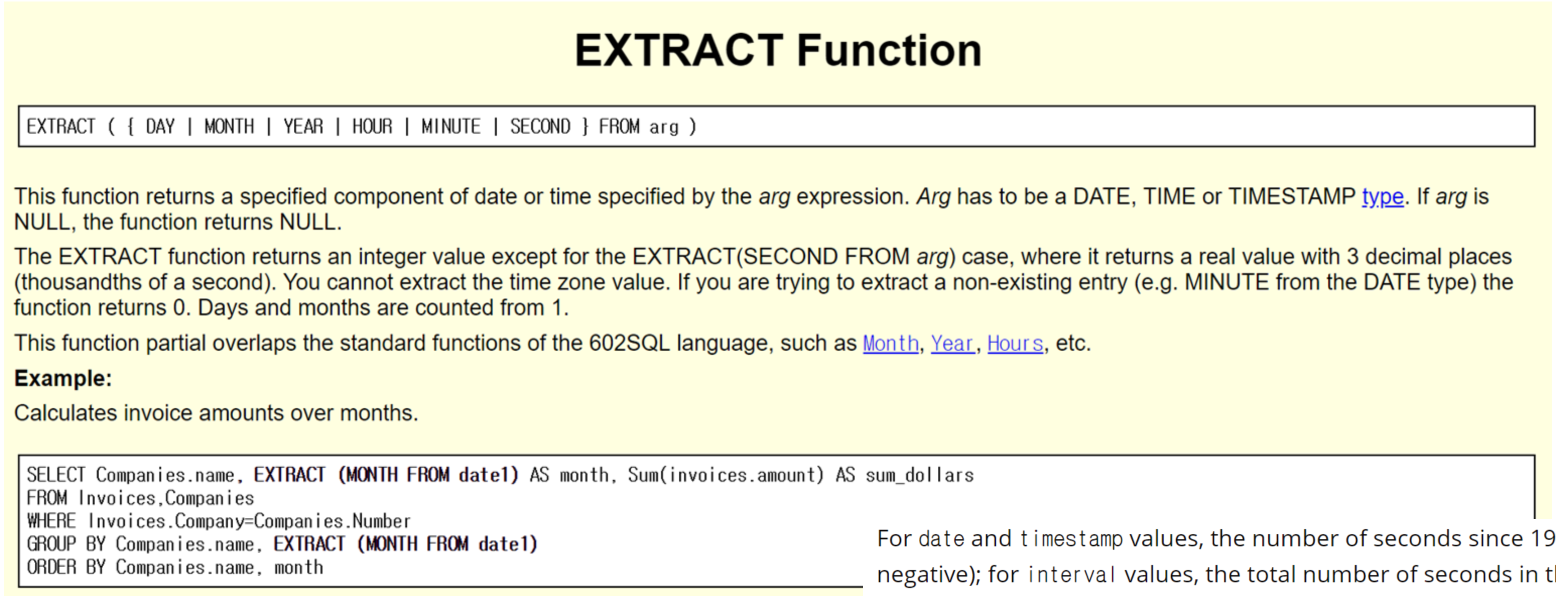
CustomerName	OrderID
Alfreds Futterkiste	*null*
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365
Around the Horn	10355
Around the Horn	10383

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

Elective Admission provides further guidance for classifying an **admission** to hospital via an **ELECTIVE ADMISSION** LIST. An **Elective Admission** is one that has been arranged in advance. It is not an emergency **admission**, a maternity**admission** or a transfer from a Hospital Bed in another Health Care Provider.

http://sql602.sourceforge.net/helpdir-en/html/HE_SQL2_EXTRACT.htm

tableone-demo.ipynb



EXTRACT Function

```
EXTRACT ( { DAY | MONTH | YEAR | HOUR | MINUTE | SECOND } FROM arg )
```

This function returns a specified component of date or time specified by the *arg* expression. *Arg* has to be a DATE, TIME or TIMESTAMP type. If *arg* is NULL, the function returns NULL.

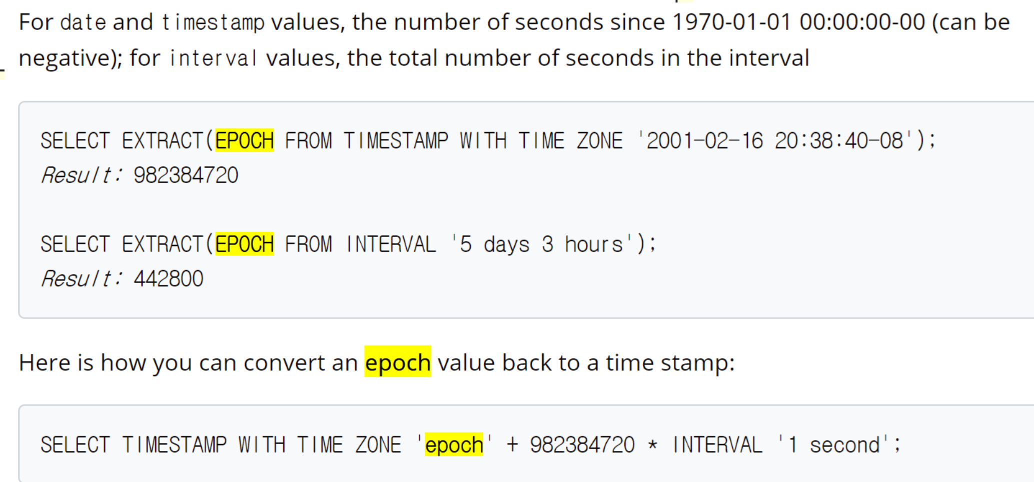
The EXTRACT function returns an integer value except for the EXTRACT(SECOND FROM *arg*) case, where it returns a real value with 3 decimal places (thousandths of a second). You cannot extract the time zone value. If you are trying to extract a non-existing entry (e.g. MINUTE from the DATE type) the function returns 0. Days and months are counted from 1.

This function partial overlaps the standard functions of the 602SQL language, such as Month, Year, Hours, etc.

Example:

Calculates invoice amounts over months.

```
SELECT Companies.name, EXTRACT (MONTH FROM date1) AS month, Sum(invoices.amount) AS sum_dollars
FROM Invoices,Companies
WHERE Invoices.Company=Companies.Number
GROUP BY Companies.name, EXTRACT (MONTH FROM date1)
ORDER BY Companies.name, month
```



For date and timestamp values, the number of seconds since 1970-01-01 00:00:00-00 (can be negative); for interval values, the total number of seconds in the interval

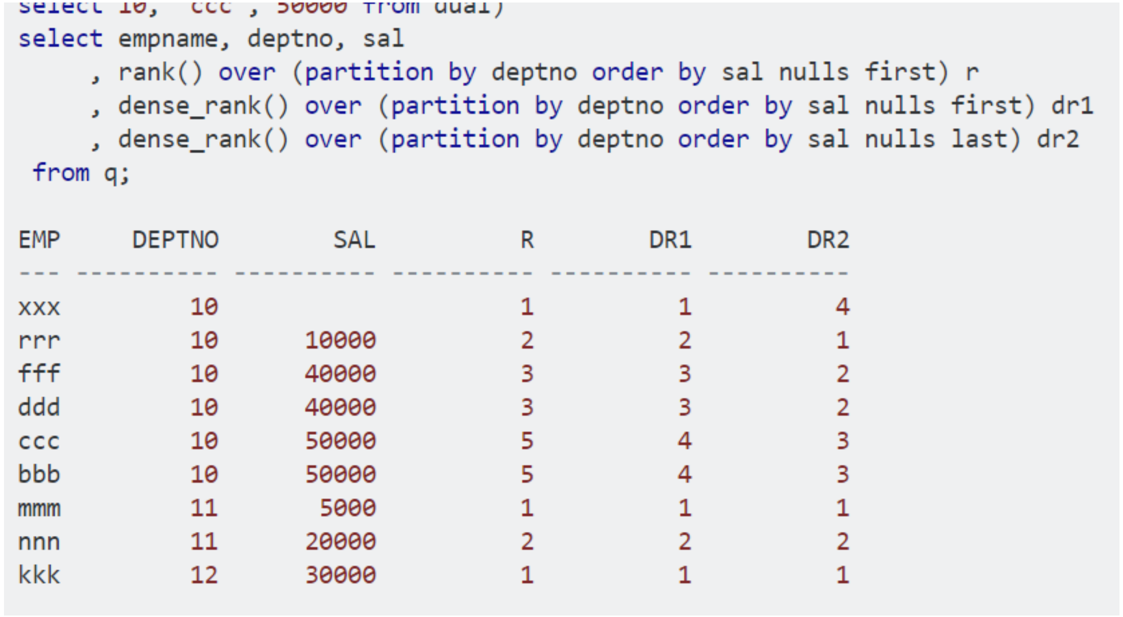
```
SELECT EXTRACT(EPOCH FROM TIMESTAMP WITH TIME ZONE '2001-02-16 20:38:40-08');
Result: 982384720

SELECT EXTRACT(EPOCH FROM INTERVAL '5 days 3 hours');
Result: 442800
```

Here is how you can convert an epoch value back to a time stamp:

```
SELECT TIMESTAMP WITH TIME ZONE 'epoch' + 982384720 * INTERVAL '1 second';
```

https://www.postgresql.org/docs/8.1/functions-datetime.html

RANK() vs. DENSE_RANK()



```
select 10, 'ccc', 50000 from dual)
select empname, deptno, sal
     , rank() over (partition by deptno order by sal nulls first) r
     , dense_rank() over (partition by deptno order by sal nulls first) dr1
     , dense_rank() over (partition by deptno order by sal nulls last) dr2
 from q;

EMP     DEPTNO        SAL          R        DR1        DR2
--- ---------- ---------- ---------- ---------- ----------
xxx         10                     1          1          4
rrr         10      10000          2          2          1
fff         10      40000          3          3          2
ddd         10      40000          3          3          2
ccc         10      50000          5          4          3
bbb         10      50000          5          4          3
mmm         11       5000          1          1          1
nnn         11      20000          2          2          2
kkk         12      30000          1          1          1
```

https://stackoverflow.com/questions/11183572/whats-the-difference-between-rank-and-dense-rank-functions-in-oracle