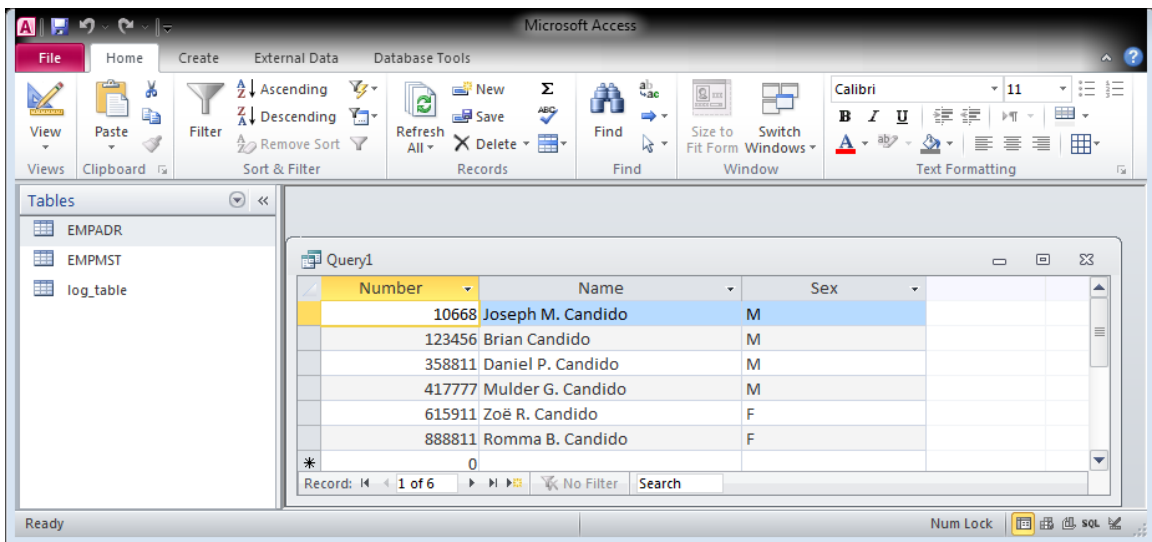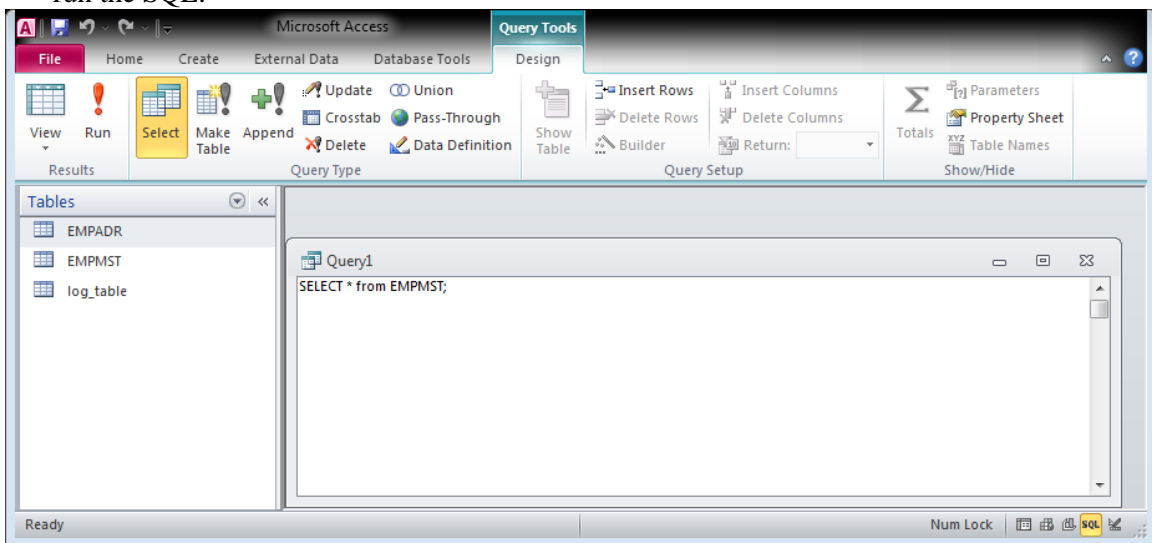# Python: SQL Select

Professor: Brian Candido

SQL (Structured Query Langauge) is a language for retrieving information out of relational databases such as Oracle, Informix, SQL Server, Access and MySQL. There are many variations of SQL and each vendor includes their own functions and extensions. This document will cover the ANSI standards that will ensure the statements will be valid regardless of the back-end database.

Note: To write and execute SQL statements in MS Access 2010 follow these directions:
1. Click on the **Create** tab.
2. Click on **Query Design** icon.
3. You will be prompted with the Show Table dialog box. Click Close.
4. Click on **View** and choose **SQL View**
5. Enter the SQL code into the SQL dialog box and click on the red exclamation mark to run the SQL.

SELECT Statement Summary:

SELECT *(fields or *)*   **Required**
FROM *library.table*   **Required**
WHERE   **Optional**
ORDER BY   **Optional**
GROUP BY   **Optional**
HAVING   **Optional**

(1) Employee Master (EMPMST): view each record in the table

select * from empmst

Number	Name	Sex
123456	Brian L. Candido	M
010668	Joseph M. Candido	M
615911	Zoë R. Candido	F
358811	Daniel P. Candido	M
417777	Mulder G. Candido	M
888811	Romma B. Candido	F

(2): Employee Address (EMPADR): view each record in the table

select * from empadr

Number	Address	City	State	ZIP
123456	17 High Ridge Road	Wilbraham	MA	01056-9552
615911	89 Cabriolet Lane	Springfield	MA	01119-1312
358811	13 Pound Ridge St	Ludlow	MA	01148-1312
417777	59 Melba Street	E. Longmeadow	MA	01122-1112
331236	806 E. 63 Street	NYC	NY	00221-1147

(3) Join 1 Link between the EMPMST and EMPADR (matched records) using traditional linking:

SELECT a.number,a.name,city,state,zip
FROM empmst as a, empadr as b
WHERE a.number = b.number

Number	Name	City	State	ZIP
123456	Brian L. Candido	Wilbraham	MA	01056-9552
615911	Zoe R. Candido	Springfield	MA	01119-1312
358811	Daniel P. Candido	Ludlow	MA	01148-1312
417777	Mulder G. Candido	E. Longmeadow	MA	01122-1112

(4) Join 2 Link between the EMPMST and EMPADR (unmatched records) using traditional linking:

SELECT number,name,sex
FROM empmst
WHERE number NOT IN (SELECT number FROM empadr)

Number	Name	Sex
10668	Joseph M. Candido	F
888811	Romma B. Candido	F

(5) Other unmatched check (EMPADR to EMPMST):

SELECT *
FROM empadr
WHERE number NOT IN (SELECT number FROM empmst)

Number	Address	City	State	ZIP
331236	806 E. 63 Street	NYC	NY	00221-1147

(6) Link between the EMPMST and EMPADR (matched records) using join syntax:

SELECT a.number,a.name,b.city,b.state,b.zip
FROM empmst AS a INNER JOIN empadr AS b
ON a.number = b.number

Number	Name	City	State	ZIP
123456	Brian L. Candido	Wilbraham	MA	01056-9552
615911	Zoe R. Candido	Springfield	MA	01119-1312
358811	Daniel P. Candido	Ludlow	MA	01148-1312
417777	Mulder G. Candido	E. Longmeadow	MA	01122-1112

(7) Link between the EMPMST and EMPADR (matched with primary) using join syntax:

SELECT a.number,a.name,b.city,b.state,b.zip
FROM empmst AS a LEFT JOIN empadr AS b
ON a.number = b.number

Number	Name	City	State	ZIP
123456	Brian L. Candido	Wilbraham	MA	01056-9552
10668	Joseph M. Candido			
615911	Zoe R. Candido	Springfield	MA	01119-1312
358811	Daniel P. Candido	Ludlow	MA	01148-1312
417777	Mulder G. Candido	E. Longmeadow	MA	01122-1112
888811	Romma B. Candido			

Note: The blanks (null) indicates there is no matching record.

(8) Link between the EMPMST and EMPADR (matched with primary) using join syntax to identify the one's with no corresponding record.

SELECT a.number,a.name,b.city,b.state,b.zip
FROM empmst AS a LEFT JOIN empadr AS b
ON a.number = b.number
WHERE b.number is null

Number	Name	City	State	ZIP
10668	Joseph M. Candido			
888811	Romma B. Candido			

(9) Link between the EMPMST and EMPADR (Cartesian one to many link) using join syntax:

SELECT a.*,b.city
FROM empmst as a ,empadr as b
WHERE a.number =   123456

Number	Name	Sex	City
123456	Brian L. Candido	M	Wilbraham
123456	Brian L. Candido	M	Springfield
123456	Brian L. Candido	M	Ludlow
123456	Brian L. Candido	M	E. Longmeadow
123456	Brian L. Candido	M	NYC

(10) Summary SQL statements (COUNT, SUM, AVG, MIN, MAX).

- Count how many rows in the table:
SELECT count(*) FROM empmst

- Count how many rows in the table:
SELECT count(*) FROM empmst

- Get the maximum name:
SELECT max(name) FROM empmst

- Get the minimum employee number:
SELECT min(number) FROM empmst

Note: You can apply COUNT, SUM, AVG, MIN and MAX on all numeric fields. You can only use COUNT, MIN and MAX on character fields.

(11) Sorting using the ORDER BY
- Ascending:
SELECT * FROM empmst ORDER BY name

- Descending:
SELECT * FROM empmst ORDER BY name DESC

(11) Using the GROUP BY

- One row for each occurrence of the group by:

SELECT sex, count(1) FROM empmst GROUP BY sex

- Alternative sort after the group by:

SELECT sex, count(1) FROM empmst GROUP BY sex ORDER BY sex DESC

Note: You don't have to use Summary functions on the GROUP BY.