SQL Reference & Resources

Select Statements

```
SELECT *
FROM table1
```

- Select all rows and columns (*) from table1

```
SELECT col1, col2
FROM table1
```

- Select columns col1 and col2 and all rows from table1

```
SELECT col1, col2
FROM table1
WHERE logical-conditions
```

- Select columns col1 and col2 and all rows from table1 where the rows meet the logical conditions. Operators: AND, OR, NOT, IN, LIKE, BETWEEN.

```
SELECT col1, sum(col2) as col2sum
FROM table1
GROUP BY col1
```

- Select the values of column col1 and the sum of col2 over the distinct values of col1. Could also use other aggregation functions: SUM, MIN, MAX, MEAN/AVG, COUNT, VAR, STDEV.

```
SELECT col1, sum(col2) as col2sum
FROM table1
GROUP BY col1
HAVING sum(col2) > 100
```

- Select the values of column col1 and the sum of col2 over the distinct values of col1, keeping only values of col1 that have a sum over 100. Could also use other aggregation functions: SUM, MIN, MAX, MEAN/AVG, COUNT, VAR, STDEV.

Joins

```
SELECT * FROM table1 INNER JOIN table2 ON join-conditions
```

- Join all rows of table1 with rows of table2 that match the join-condition (typically identical keys).

```
SELECT * FROM table1 LEFT JOIN table2 ON join-conditions
```

- Join all rows of table1 with rows of table2 that match the join-condition (typically identical keys). Result contains all rows of table1; rows that do not match the join-condition have missing/null values for the columns from table2

```
SELECT * FROM table1 RIGHT JOIN table2 ON join-conditions
```

- Join all rows of table1 with rows of table2 that match the join-condition (typically identical keys). Result contains all rows of table2; rows that do not match the join-condition have missing/null values for the columns from table1

Online References

- SAS Online Documentation
  - http://support.sas.com/onlinedoc/913/docMainpage.jsp
  - Base SAS → Base SAS Procedures Guide → SAS SQL Procedure User's Guide
- MS Access SQL Documentation
  - http://office.microsoft.com/en-us/access-help/CH010072899.aspx
- SQL Tutorial
  - http://www.sqltutorial.org
  - SQL Cheat Sheet - http://www.sqltutorial.org/sql-cheat-sheet.aspx

Books

- O'Reilly Books (http://shop.oreilly.com)
  - Learning SQL, 2nd Edition
  - SQL In a Nutshell, 3rd Edition
  - SQL Cookbook, 1st Edition
  - SQL Pocket Guide, 3rd Edition
- SAS Publishing (http://support.sas.com/publishing/)
  - PROC SQL: Beyond the Basics Using SAS
  - PROC SQL By Example: Using SQL Within SAS

SQL CHEAT SHEET

http://www.sqltutorial.org

SQL SELECT STATEMENTS

```
SELECT * FROM tbl
```

Select all rows and columns from table tbl

```
SELECT c1,c2 FROM tbl
```

Select column c1, c2 and all rows from table tbl

```
SELECT c1,c2 FROM tbl
WHERE conditions
ORDER BY c1 ASC, c2 DESC
```

Select columns c1, c2 with where conditions and from table tbl order result by column c1 in ascending order and c2 in descending order

```
SELECT DISTINCT c1, c2
FROM tbl
```

Select distinct rows by columns c1 and c2 from table tbl.

```
SELECT c1, aggregate(expr)
FROM tbl
GROUP BY c1
```

Select column c1 and use aggregate function on expression expr, group columns by column c1.

```
SELECT c1, aggregate(expr) AS c2
FROM tbl
GROUP BY c1
HAVING c2 > v
```

Select column c1 and c2 as column alias of the result of aggregate function on expr. Filter group of records with c2 greater than value v

SQL UPDATE TABLE

```
INSERT INTO tbl(c1,c2,...)
VALUES(v1,v2...)
```

Insert data into table tbl

```
INSERT INTO tbl(c1,c2,...)
SELECT c1,c2.. FROM tbl2
WHERE conditions
```

Insert data from tbl2 into tbl

```
UPDATE t
SET c1 = v1, c2 = v2...
WHERE conditions
```

Update data in table tbl

```
DELETE FROM tbl
WHERE conditions
```

Delete records from table tbl based on WHERE conditions.

```
TRUNCATE TABLE tbl
```

Drop table tbl and re-create it, all data is lost

SQL TABLE STATEMENTS

```
CREATE TABLE tbl(
        c1 datatype(length)
        c2 datatype(length)
        ...
        PRIMARY KEY(c1)
)
```

Create table tbl with primary key is c1

```
DROP TABLE tbl
```

Remove table tbl from database.

```
ALTER TABLE tbl
ADD COLUMN c1 datatype(length)
```

Add column c1 to table tbl

```
ALTER TABLE tbl
DROP COLUMN c1
```

Drop column c1 from table tbl

SQL JOIN STATEMENTS

```
SELECT * FROM tbl1
INNER JOIN tbl2 ON join-conditions
```

Inner join table tbl1 with tbl2 based on join-conditions.

```
SELECT * FROM tbl1
LEFT JOIN tbl2 ON join-conditions
```

Left join table tbl1 with tbl2 based on join-conditions.

```
SELECT * FROM tbl1
RIGHT JOIN tbl2 ON join-conditions
```

Right join table tbl1 with tbl2 based on join-conditions.

```
SELECT * FROM tbl1
RIGHT JOIN tbl2 ON join-conditions
```

Full outer join table tbl1 with tbl2 based on join-conditions.

```
/* Sample code for TAIR 2012 Presentation */

libname tair "C:\Documents and Settings\prhodes\Desktop\TAIR 2012\data";
run;


proc sql;   /* Example 1:  Basic Query */
    select *
    from tair.students;
quit;

proc sql;   /* Example 2:  Subsetting */
    select *
    from tair.students
    where home_state ne 'TX';
quit;

proc sql;   /* Example 3:  Inner Join */
    select a.id, a.home_state, b.class
    from tair.students as a,
         tair.acad_info as b
    where a.id = b.id and
            home_state ne 'TX';
quit;

proc sql;   /* Example 4:  Left Join */
    select a.id, a.gender, b.course
    from tair.students as a
        left join tair.schedule as b
    on a.id=b.id
    where a.home_state<>'TX';
quit;

proc sql;   /* Example 5:  Basic Report */
    select home_state as state label='State',
        count(distinct id) as number label='Number'
    from tair.students
    group by home_state;
quit;

proc sql;   /* Example 6:  Basic Report 2 */
    select a.id, b.class,
           sum(c.credit_hrs) as total_hours
    from tair.students as a,
         tair.acad_info as b,
         tair.schedule as c
    where a.id=b.id and
          b.id=c.id
    group by a.id, b.class
    having sum(c.credit_hrs)>18
    order by class;
quit;
```