MASTERNOTES

DATA MANAGMENT

CATANNA, FALL 2014

**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -**

**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -**

***CHAPTER 3***

**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -**

**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -**

***CHAPTER 4***

* **JOIN** lets you combine columns from 2 or more tables into a single result set
* Structure:
  + **FROM** [specify tables to join]
    - Ex: **FROM** table1 **JOIN** table2
  + **ON** [join conditions]
    - Specify a column in each table involved in join
    - Use any comparison operator (=, **LIKE**, **NOT LIKE**, etc.)
  + most common type
  + only rows that satisfy join condition are included in result set

1. **INNER JOIN**
   * most common type
   * only rows that satisfy join condition are included in result set
   * you can leave out the "**INNER**"; a **JOIN** is an inner join by default
   * **CASE1:** *aliases* for tables in **FROM** without **AS**
     + Ex: **FROM** table1 alias1 **JOIN** table2 alias2
   * **CASE2:** tables from *differernt schemas*
     + log on as user that has access permisions
     + to grant permissions:
       - Ex: **GRANT SELECT ON** tableNameInSchema1 **TO** schema2
     + then use in select statement, making sure to prefix schema name to table name
       - Ex: **JOIN** schema1.tablename
   * **CASE3:** 2 or more join conditions
     + either use **AND**/**OR** operations in the **FROM**
     + or put second condition in **WHERE** (easier to read)
   * **CASE4:** more than 2 tables
     + each join is based on the relationship between primary key of one table and a foreign key of the other table
2. **OUTER JOIN**
   * Returns all rows that match join condition AND returns all rows from one or both tables, regardless of join condition
   * Must specifiy keyword: **LEFT**/**RIGHT**/**FULL**
   * you can leave out the "**OUTER**" or you can include it
   * **CASE1: LEFT OUTER JOIN**
     + Includes all rows from first table
     + If no matches found in second table, second table returns null for that row
     + Ex: **FROM** table1 alias1 **JOIN** table2 alias2
   * **CASE1: RIGHT OUTER JOIN**
     + Includes all rows fromsecond table
     + try to avoid using right join; switch the table order in left join
     + Ex: **FROM** table1 alias1 **JOIN** table2 alias2
   * **CASE1: LEFT OUTER JOIN**
     + Includes all rows from all tables
     + Ex: **FROM** table1 alias1 **JOIN** table2 alias2
   * **CASE2:** tables from *differernt schemas*
3. log on as user that has access permisio
4. sdf
5. sdf
6. sdf

a) hello

b) hello