1. *How do you display letters in uppercase in Oracle, Access, and SQL Server? How do you display letters in lowercase in Oracle, Access, and SQL Server?*
   1. ORACLE: UPPER(*string)*
   2. ACCESS: UCASE(*string)*
   3. SQL SERVER: UPPER(*string)*
2. *How do you round a number to a specific number of decimal places in Oracle, Access, and SQL Server? How do you remove everything to the right of the decimal place in Oracle, Access, and SQL Server?*
   1. ORACLE: ROUND( number [, decimal\_places] )
   2. ACCESS: Round ( expression, [ decimal\_places ] )
   3. SQL SERVER: ROUND( *number, decimal\_places [, operation ]* )
   4. ORACLE: RTRIM( string1 [, trim\_string ] )
   5. ACCESS: RTrim ( text )
   6. SQL SERVER: RTRIM( string )
3. *How do you add months to a date in Oracle, Access, and SQL Server? How do you add days to a date? How would you find the number of days between two dates?*
   1. ORACLE: ADD\_MONTHS( date1, number\_months )
   2. ACCESS: DateAdd ( interval, number, date )
   3. SQL SERVER: DATEADD( interval, number, date )
   4. ORACLE: SELECT ( SYSDATE + N ) FROM TABLE;
   5. ACCESS: DateAdd ( interval, number, date )
   6. SQL SERVER: DATEADD( interval, number, date )
   7. ORACLE: select date '2000-01-02' - date '2000-01-01' as dateDiff
   8. ACCESS: DateDiff ( interval, date1, date2, [firstdayofweek], [firstweekofyear])
   9. SQL SERVER: DATEDIFF( interval, date1, date2 )
4. *How do you obtain today’s date in Oracle, Access, and SQL Server?* 
   1. ORACLE: SYSDATE
   2. ACCESS: Date ()
   3. SQL SERVER: GETDATE ( )
5. *How do you concatenate values in character columns in Oracle, Access, and SQL Server?*
   1. ORACLE: ||
   2. ACCESS: &
   3. SQL SERVER: +
6. *Which function deletes extra spaces at the end of a value?* 
   1. RTRIM()
7. *What are stored procedures? What purpose do they serve?* 
   1. Stored Procedures are blocks of queries, sometimes encapsulated by blocks of logic, that are named and saved to the database. They are basically user defined functions that can be called to avoid retyping or executed within other Stored Procedures or even themselves to recursively execute queries or commands.
8. *In which portion of a PL/SQL procedure do you embed SQL commands?*
   1. SQL queries are placed in the Executable block of a procedure, after the Declarative block
9. *Where do you declare variables in PL/SQL procedures?*
   1. Variables are declared in the Declarative block of a procedure, before the Execution block.
10. *In PL/SQL, how do you assign variables the same type as a column in the database?*
    1. Use %TYPE when assigning the datatype
11. *How do you place the results of a SELECT command into variables in PL/SQL?*
    1. SELECT *column1, column2* INTO *variable1, variable2*
12. *Can you use INSERT, UPDATE, or DELETE commands that affect more than one row in PL/SQL procedures?*
    1. UPDATE and DELETE can affect multiple rows.
13. *How do you use a SELECT command that retrieves more than one row in a PL/SQL procedure?*
    1. You can implement a cursor to iterate over a return with multiple rows
14. *Which PL/SQL command activates a cursor?*
    1. CURSOR *cursor\_name* IS SELECT*\_statement;*

OPEN cursor\_name;

1. *Which PL/SQL command selects the next row in a cursor?* 
   1. FETCH *cursorName* INTO *variable1, variable2*
2. *Which PL/SQL command deactivates a cursor?* 
   1. CLOSE *cursorName;*
3. *How do you use SQL commands in Access?* 
   1. SQL commands are passed as string-literals to the DoCmd.RunSQL object within Visual Basic
4. *How do you process a collection of rows retrieved by a SELECT command in Access?* 
   1. Instantiate an ADODB.Recordset object
   2. Open the recordset by passing the SQL query to it
   3. Iterate through the results with a DO UNTIL end-of-file loop

1. *How do you move to the next record in a recordset in Access?* 
   1. The ADODB.Recordset has a method called .MoveNext that moves to the next record in the set
2. *What are triggers? What purpose do they serve?* 
   1. Triggers is SQL’s way of handling user defined events. If something changes on one table, that may need to affect the data of another table. Triggers listen for the events they’re assigned to and execute commands when that trigger is envoked.
3. *What is the purpose of the INSERTED and DELETED tables in SQL Server?*
   1. The INSERTED and DELETED tables are temporary, in-memory only, tables SQL Server creates upon an INSERT/UPDATE/DELETE Trigger. The INSERTED table holds the new information that a user is attempting to insert into a table and the DELETED table holds the information that a user is attempting to delete (or the old information that is written over with UPDATE). They allow you to reference the data so you can store them in variables and apply functions to the data in Triggers.