**Part 1 - Inner Joins Pg. 206 – 209**

**Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated**  
*Type in the queries in the SQL box from steps 2, 3, and 4:*

*Type in the query that is under the first tip on page 209:*

Text

Description automatically generated  
  
1. What does a join do in SQL? Defend your answer with an example:

*A join in SQL allows for the results of a query to be combined from two (or more) tables where certain parameters are matched. An example from our database would be:*

SELECT f.name,

FROM `forums` AS f

INNER JOIN `messages` AS m

USING (forum\_id);

*This query would return the names of all forums which contain messages in the messages table.*

2. What does an inner join do in SQL? Defend your answer with an example.

*An inner join returns only the columns from two related tables where a match is found in both tables, such as:*

SELECT m.subject,

FROM `users` AS u

INNER JOIN `messages` AS m

USING (user\_id)

WHERE u.username = ‘tim’;

*This would return all results from the messages column that contained the same user\_id that matches the username column in the users table with the username “tim”.*

**Part 2 - Outer Joins and Joining 3 or more tables Pgs. 210 – 215**

Type in the query from step 2 on page 211:

Text

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generatedType in the query from step 3 on page 212:

Text

Description automatically generated  
Type in the query from step 2 on page 214:

Text

Description automatically generatedText

Description automatically generated6. Type in the queries from step 3 and 4 on page 215:

1. Compare and contrast the types of outer joins: left, right, and full. What do they  
do? How do they differ? What solutions do they provide for SQL queries? Defend  
your answer with examples.

*A left outer join returns all the values from the first table specified in a query along with the values from the second table which match the query parameter. As an example of a left join between the users table and the messages table using the user\_id parameter would return all the users (including users that had not posted any messages), along with the ids of all users that had posted messages.*

*A right join is just the opposite of the left, returning the entirety of the values from the second table, along with the matching values from the first table. A right join of the same parameters would return all the messages, but only the users which had posted messages.*

*A full join returns all the values from both tables, including those which don’t match the join parameter. This query is usually not necessary nor directly supported in SQL but could be achieved using a union statement with the previous left and right joins and would return all users and messages.*

2. Define self-joins and discuss how they are implemented.

*A self-join allows a table to be joined with itself by using multiple aliases for the table in a join statement.*

3. What benefits do you get from using self-joins?

*Self-joins allow subsets of data within a table to be queried, such as which messages are replies to another message.*

**Part 3 - Grouping Selected Results Pgs. 216 - 219**

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated with medium confidenceGraphical user interface, text, application

Description automatically generatedType in the queries from steps 2,3,4, and 5 on pages 218 and 219:

Discuss the functions in Table 7.1 on page 216. Define them and show an example  
of each using SQL:

AVG() The average of the values in a column

*Graphical user interface, text

Description automatically generatedThe AVG() function aggregates all values in a column and returns the mean value.*

COUNT() The number of values in a column

*Graphical user interface, text

Description automatically generatedThe COUNT() function returns the total number of rows in a column.*

GROUP\_CONCAT() The concatenation of a column’s values

*Graphical user interface, text

Description automatically generatedThe GROUP\_CONCAT() function returns the values (if there are more than one) of the column in a comma-separated list.*

MAX() The largest value in a column

*Graphical user interface, text

Description automatically generatedThe MAX() function finds and returns the largest value in the column.*

MIN() The smallest value in a column

*Graphical user interface, text

Description automatically generatedThe MIN() function finds and returns the smallest value in the column.*

SUM() The sum of all the values in a column

Graphical user interface, text

Description automatically generated*The SUM() function tallies the values of all the entries in a column and returns their total combined value.*

**Part 4 - Advanced Selections Pgs. 220 – 223**

Text

Description automatically generatedText

Description automatically generated  
Type in the queries from steps 2 and 3 on page 222:

Text

Description automatically generated  
Type in the query on step 4 on page 223:

1. Define COALESCE, GREATEST, LEAST, and CONCAT functions and show an  
example of each in SQL.

*Text

Description automatically generated with medium confidenceCOALESCE: The COALESCE() function returns the first non-null value found in a column.*

*Text

Description automatically generatedGREATEST: The GREATEST() function returns the largest value of any column specified in the same row, or from a given list of values.*

*Text

Description automatically generatedLEAST: The LEAST() function returns the smallest value of any column specified in the same row, or from a given list of values.*

*A picture containing text

Description automatically generatedCONCAT: The CONCAT() function takes the values of all parameters given to it a returns them in a sequential string.*

2. How would you use SQL to select dates from the past until now?

*To select all dates from the past until now a SELECT statement would need to be used on a table with a DATETIME column and an ORDER BY <column name> ASC statement could be appended to the query to return all of the values from the in order from the oldest to the most recent.*

**Part 5 - Full Text Searches Pgs. 224 – 231**

Text

Description automatically generated  
Type the query from step 2 on page 225:

Graphical user interface, text

Description automatically generatedType the queries from steps 3 and 4 on pages 225 and 226:

Type the query from step 3 on page 227:  
Text

Description automatically generated with medium confidence

Text

Description automatically generatedText

Description automatically generatedType the queries from steps 2 and 3 on pages 230 and 231.

Text

Description automatically generated  
Type the query from step 4 on page 231:

1. Compare and contrast full text searches with the searches with the like keyword  
from Lab 6. What are some of the main benefits that you get from using full text  
searching? When is it appropriate to use full text searching?

*A keyword search allows for the database to seek out single words or short strings of characters, while full text searches allow for complex case-insensitive multi-word strings to be searched for amongst columns in a database that have been properly indexed. This allows for a much more ‘google-like’ search functionality, at the cost of greater resource needs. Usually this would be most appropriate for columns that users would want to search for particular items of interest in a large pool of data.*

**Part 6 - Optimizing Queries Pgs. 232 - 235**

A computer screen capture

Description automatically generated with low confidenceType the query from step 2 on page 233:

1. How would you confirm that the query is, in fact, optimized? Discuss how you  
would look at the results to determine that. Your response should be limited to a  
maximum of 2 paragraphs, no more than 10 sentences.

**Part 7 - Performing Transactions Pgs. 236 – 238**

Text

Description automatically generatedText

Description automatically generated  
1. Type in the queries from steps 2 through 9 on pages 237 and 238:

Text

Description automatically generated  
1. What are transactions and why are they utilized in SQL?

2. What does autocommit option do if it is set to 0?

3. What are save points and how are they implemented in SQL?

**Part 8 - Database Encryption Pgs. 239 – 241**

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated  
1. Type the queries from steps 2 through 5 on pages 240 and 241:

1. Define AES Encryption.

*AES Encryption allows information to be stored in a database in a coded, unreadable format. Along with a salt that is stored and an encryption key, the data stored can be returned to plain text, unlike a hashing algorithm.*

2. How many bytes of encryption does it use?

*AES utilizes either 128, 192 or 256 bytes, depending on the specification used.*

3. Who invented it?

*Joan Daemen and Vincent Rijmen invented AES Encryption.*

4. Why do you think we use AES Encryption in this lab but SHA1 for the password in  
previous labs?

5. Compare and contrast SHA1 with AES Encryption.