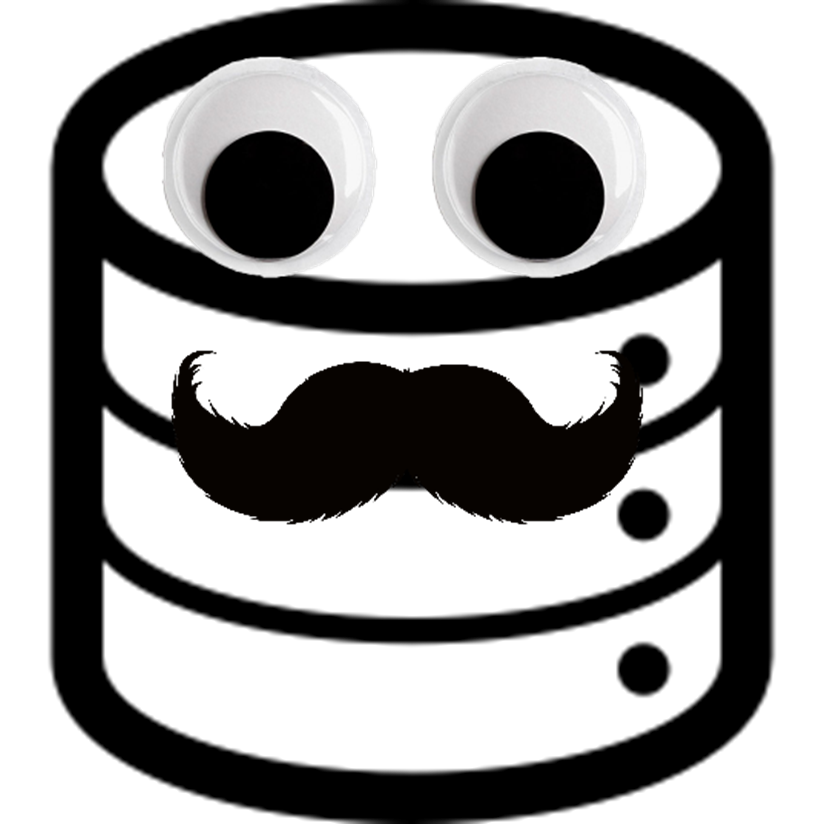
*WELCOME TO CLASS*



*SQL for Smart Folks\* Intermediate Bootcamp*

Instructor: [Michael Valeri](https://www.linkedin.com/in/mvaleri/)

*\*YOU – Common People*

**0. Pre-Work – Download PG Admin and Getting Curriculum**

|  |
| --- |
| ***\*GETTING STARTED GUIDE\****  *Step 1:* [*Download PG Admin 4*](https://www.pgadmin.org/download/)*. Select OS from list. If trouble opening on Mac after download, see troubleshoot guide* [*here*](https://www.pgadmin.org/faq/)*.*    *<Windows/MAC 🡪 Select latest version from list 🡪 2nd Option .EXE/.DMG*    *Step 2: Find SQL Bootcamp Curriculum* [*here*](https://github.com/mvaleri12/valeri-analytics-sql-bootcamp)  *<Open Link above 🡪 Download Zip>*  *<Find Zip File in Downloads Folder called valeri-analytics-sql-bootcamp-master>*  *<Open Zip File 🡪 Drag folder onto Desktop so you can find later>*    *Step 3: Connect to AWS server on PG Admin 4; credentials on Page 6*  *Step 4: Start brainstorming questions on introduction example, Page 3* |

1. **REVIEW – Basic SQL Keywords**

**WITH**

|  |  |  |
| --- | --- | --- |
| **SEQ.** | **KEYWORD** | **WHAT IT DOES** |
| 1 | SELECT |  |
| 2 | FROM |  |
| 3 | WHERE |  |
| 4 | GROUP BY |  |
| 5 | HAVING |  |
| 6 | ORDER BY |  |
| 7 | LIMIT |  |

1. **REVIEW – Opening Activity, Answer This Question**

*“The governor of Iowa is trying to determine how many dollars*

*were spent on over 100 proof alcohol products in Iowa during 2015 alone;*

*particularly within counties that have a population count under 20,000 people.*

*He'd like to understand these sales dollars by county and population size. Order*

*the results by highest sales to lowest sales.”*

***STEPS: Draw the Final Table, Build the Shell, and Architect the Final Query***

**Step 1: Draw the Final Table Output**

*Result Output Column Headers (SELECT)*

|  |  |  |
| --- | --- | --- |
| **(1)** | **(2)** | **(3)** |

FILTERS RUNNING IN BACKGROUND OF QUERY (WHERE)

|  |  |  |
| --- | --- | --- |
| **(1)** | **(2)** | **(3)** |

**Step 2: Build the “Shell” for the Query**

|  |  |  |
| --- | --- | --- |
| **SEQ.** | **KEYWORD** | **WHAT COLUMNS GO HERE** |
| 1 | SELECT |  |
| 2 | FROM |  |
| 3 | WHERE |  |
| 4 | GROUP BY |  |
| 6 | ORDER BY |  |

**Step 3: Architect the query using the hinted column names in the text document**

|  |  |  |
| --- | --- | --- |
| **Table A: Sales** | **Table B: Counties** | **Table C: Products** |
| TOTAL – Total dollars spent by customer  COUNTY – Name of county  DATE – Date product was sold on  ITEM – Unique ID of item | COUNTY – Name of county  POPULATION – Population of county | ITEM\_NO – Unique ID of item  PROOF (text) – Proof of alcohol content |

1. **Creating Conditional Statements (Case Statements)**

*Table COLORS*



1. When do you use a CASE statement?
2. What is a CASE statement comparable to in Excel?
3. What is the syntax of a CASE statement? Write it here.
4. What can you pair CASE statements with?
5. **Window Functions (Example Ranking Functions in SQL)**

*Table ALBUMS\_LITE*



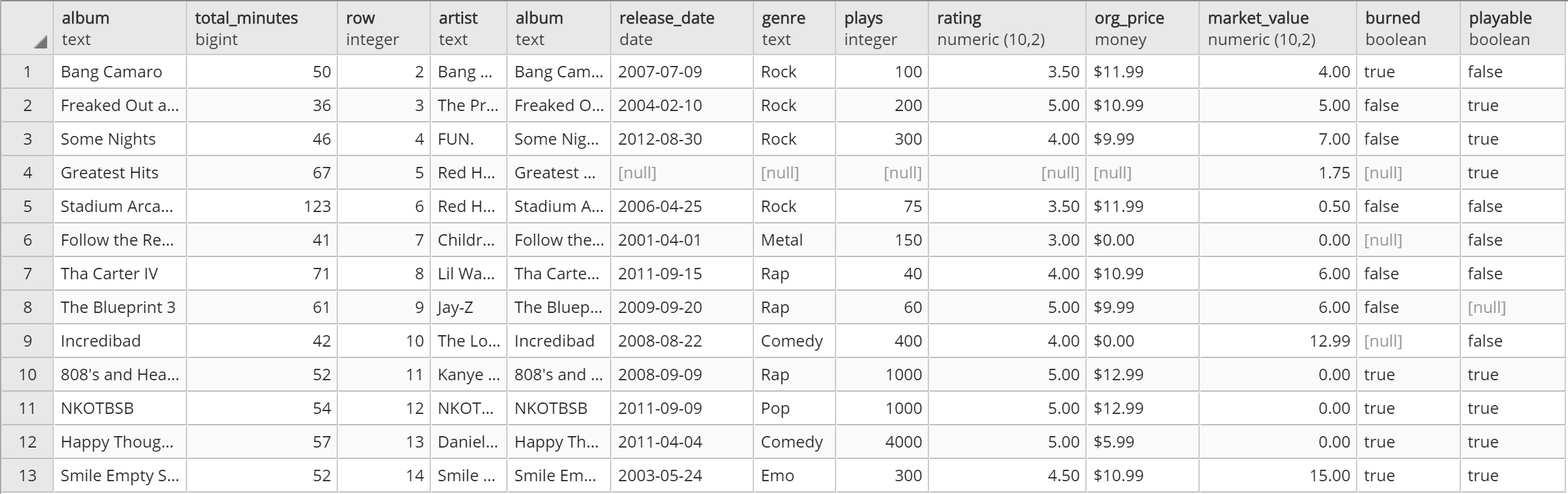
|  |  |  |
| --- | --- | --- |
| No Partitions | Albums Ranked by Plays (1)  RANK() OVER(ORDER BY PLAYS DESC) | Artist Ranked by Plays (2)  RANK() OVER(ORDER BY SUM(PLAYS) DESC) |
| Has Partitions | Albums Ranked Within Genre By Plays (3)  RANK() OVER(PARTITION BY GENRE ORDER BY PLAYS DESC) | Artists Ranked Within Genre By Plays  (4)  RANK() OVER(PARTITION BY GENRE ORDER BY SUM(PLAYS) DESC) |
|  | *No Δ in Table Granularity* | *Δ in Table Granularity* |

|  |  |
| --- | --- |
| What is a Window Function? | What do all Window Functions Have? |

1. **Having, Subqueries and CTEs**

*“I want to see a list of Rap Albums that are at least 60 minutes in length. Which is about the amount of time I like to hang out with my bud.”* – Dr. Dre

*Table ALBUMS*



1. How would you write this query, using the above table?
2. Where do we find Total Minutes? What’s the problem?
3. What 3 steps do we need to take to write the query for Dr. Dre?

|  |  |  |
| --- | --- | --- |
| (1) | (2) | (3) |

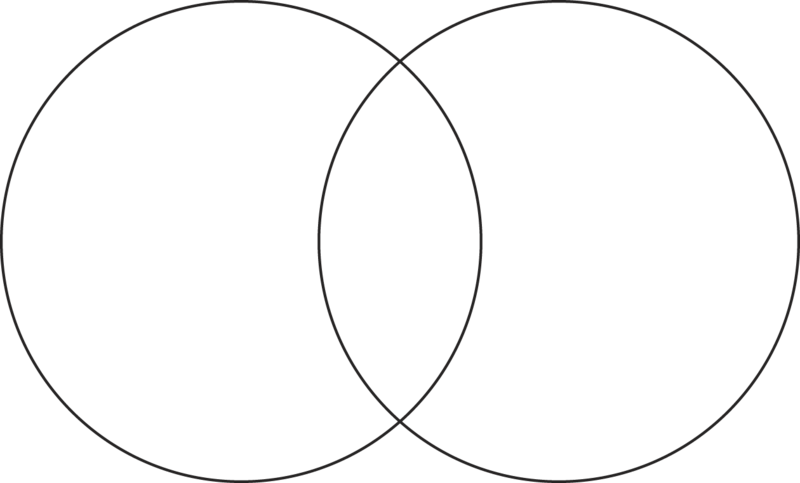
Step 1 & 2: Write the queries

QUERY\_A – *Pulls a list of Rap Albums*

QUERY\_B – *Pulls a list of Albums at least 60 minutes in length –* ***Keyword 5!***

Step 3a: JOIN these queries together

|  |  |
| --- | --- |
| **QUERY\_A** | **QUERY\_B** |
|  |  |



COLUMN TO JOIN ON?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Step 3b: Choose an Approach

|  |  |
| --- | --- |
| 1. **SUBQUERY**   ***Definition*** | 1. **CTE (WITH statement… ) *Definition*** |
| A subquery is a SELECT statement that is nested within another SELECT statement | A CTE is a temporary result set that you can reference within another SELECT statement |

1. Why do you use CTEs or Subqueries?
2. **Closing Activity, Build This Table**

***STEPS: Draw the Final Table, Build the Shell, and Architect the Final Query***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **County** | **Total Sales 2015** | **Total Sales**  **2014** | **Under 20K People?** | **Number 1 Product Name (2015)** | **Number 1 Product Sales**  **(2015)** |

**QUERY\_A QUERY\_B**

**Query\_A**

**Tables: SALES and COUNTIES**

**Columns: TOTAL, POPULATION, DATE, COUNTY**

**Query\_B**

**Tables: SALES**

**Columns: TOTAL, DESCRIPTION, DATE, COUNTY**