Wei Cheng Wu
BA Fall 2024

Beazley's Last Statement

1) A first SQL Query

- SELECT * FROM executions LIMIT 3
- SELECT TOP 3 * FROM tx_deathrow

2) The select block

- SELECT first_name, last_name, last_statement
  FROM executions
  LIMIT 3
- SELECT TOP 3 first_name, last_name, last_statement
  FROM tx_deathrow

3) The FROM block

- SELECT first_name FROM executions LIMIT 3
- SELECT TOP 3 first_name
  FROM tx_deathrow

4) Modify the query to divide 50 and 51 by 2

- SELECT 50/2, 51/2
- SELECT 50/2, 51.0/2

5) The WHERE block

- SELECT first_name, last_name, ex_age
  FROM executions
  WHERE ex_age<=25
- SELECT first_name, last_name, Age_at_Execution
  FROM tx_deathrow
  WHERE Age_at_Execution<=25

6) Modify the query to find the result for Raymond Landry

- SELECT first_name, last_name, ex_number
  FROM executions
  WHERE first_name LIKE 'Raymond%'
  AND last_name LIKE 'Landry%'
- SELECT first_name, last_name, TDCJ_Number
  FROM tx_deathrow
  WHERE first_name LIKE 'Raymond%'
  AND last_name LIKE 'Landry%'

7) Insert a pair of parentheses so that this statement returns 0

- SELECT 0 AND (0 OR 1)
- SELECT IIF(0=1 AND (0=1 OR 1=1),1,0)

8) Find napoleon beazley's last statement

- SELECT last_statement
  FROM executions
  WHERE first_name= "Napoleon"
  AND last_name= "Beazley"
- SELECT last_statement
  FROM tx_deathrow
  WHERE first_name= "Napoleon"
  AND last_name= "Beazley"

<u>Claims of Innocence</u>

1) The COUNT Function

- SELECT COUNT(last_statement)
  FROM executions
- SELECT COUNT(last_statement)
  FROM tx_deathrow

2) Nulls

- SELECT (0 IS NOT NULL) AND ('' IS NOT NULL)
- SELECT
  (CASE WHEN 0 IS NOT NULL THEN '0 is NOT NULL' ELSE '0 is NULL' END)

3) Find the total number of executions in the dataset

- SELECT COUNT(ex_number)
  FROM executions
- SELECT COUNT(TDCJ_number)
  FROM tx_deathrow

4) Variations on COUNT

- SELECT COUNT(*) FROM executions
- SELECT COUNT(*) FROM tx_deathrow

Part2:

- SELECT
  COUNT(CASE WHEN county='Harris' THEN 1
  ELSE NULL END),
  COUNT(CASE WHEN county='Bexar' THEN 1
  ELSE NULL END)
  FROM executions

- SELECT SUM(CASE WHEN county= “Harris” THEN 1 ELSE 0 END), AS Harris_count
  SUM(CASE WHEN county= “Bexar” THEN 1 ELSE 0 END), AS Bexar_count
  FROM tx_deathrow

5) Find how many inmates were over the age of 50 at execution time

- SELECT COUNT(*)
  FROM executions
  WHERE ex_age>50
- SELECT COUNT(*)
  FROM tx_deathrow
  WHERE Age_at_Execution>50

6) Find the number of inmates who have declined to give a last statement

- SELECT COUNT(*)
  FROM executions
  WHERE last_statement is NULL
- SELECT COUNT(*)
  FROM tx_deathrow
  WHERE last_statement is NULL

7) Find the minimum, maximum, and average age of inmates at the time of execution

- SELECT MIN(ex_age), MAX(ex_age), AVG(ex_age)
  FROM executions
- SELECT MIN(Age_at_execution) AS Min_Age , MAX(Age_at_Execution) AS Max_Age, AVG(Age_at_Execution) AS Avg_Age
  FROM tx_deathrow

8) Find the average length of last statements in the dataset

- SELECT AVG(LENGTH(last_statement))
  FROM executions
- SELECT AVG(LEN(last_statement)) AS Avg_Length_last_statement
  FROM tx_deathrow

9) List all the counties in the dataset without duplication

- SELECT DISTINCT county
  FROM executions
- SELECT DISTINCT county
  FROM tx_deathrow

10) A strange query

- SELECT first_name, COUNT(*)
  FROM executions
- SELECT first_name, COUNT(*) AS COUNT
  FROM tx_deathrow
  GROUP BY first_name

11) Conclusion and Recap

- SELECT 1.0 * COUNT(CASE WHEN last_statement LIKE "%innocent" THEN 1 ELSE NULL END) / COUNT(*)
  FROM executions
- SELECT 1.0 * COUNT(CASE WHEN last_statement LIKE "%innocent" THEN 1 ELSE NULL END) / COUNT(*)
  FROM tx_deathrow

<u>The Long Tail</u>

1) The GROUP BY block

- SELECT county, COUNT(*) AS county_executions
  FROM executions
  GROUP BY county
- SELECT county, COUNT(*) AS county_executions
  FROM tx_deathrow
  GROUP BY county

2) This query counts the executions with and without last statements

- SELECT last_statement IS NOT NULL AS has_last_statement, county, COUNT(*)
  FROM executions
  GROUP BY has_last_statement, county
- SELECT CASE WHEN last_statement IS NOT NULL THEN 1 ELSE 0 END AS has_last_statement, COUNT(*) AS statement_count
  FROM tx_deathrow
  GROUP BY CASE WHEN last_statement IS NOT NULL THEN 1 ELSE 0 END

3) The HAVING Block

- SELECT county, COUNT(*)
FROM executions
WHERE ex_age>=50
GROUP BY county
- SELECT county, COUNT(*) AS COUNT
  FROM tx_deathrow
  WHERE Age_at_Execution>=50
  GROUP BY county

4) List the countries in which more than 2 inmates aged 50 or older have been executed

- SELECT county, COUNT(*)
  FROM executions
  WHERE ex_age>=50
  GROUP BY county
  HAVING COUNT(*)>2
- SELECT county, COUNT(*) AS count
  FROM tx_deathrow
  WHERE Age_at_Execution>=50
  GROUP BY county
  HAVING COUNT(*)>2

5) List all the distinct counties in the dataset

- SELECT county
  FROM executions
 GROUP BY county
- SELECT county
  FROM tx_deathrow
  GROUP BY county

6) Find the first and last name of the inmate with the longest last statement

- SELECT first_name, last_name
  FROM executions

WHERE LENGTH(last_statement) = (SELECT MAX(LENGTH(last_statement)) FROM executions)
- SELECT first_name, last_name
FROM tx_deathrow
WHERE LEN(last_statement) = (SELECT MAX(LEN(last_statement)) FROM tx_deathrow)

7) Insert the count of all rows query to find the percentage of executions from each county
- SELECT county, 100.0 * COUNT(*) / (SELECT COUNT(*) FROM executions) AS percentage
FROM executions
GROUP BY county
ORDER BY percentage DESC
- SELECT county, 100.0 * COUNT(*) / (SELECT COUNT(*) FROM tx_deathrow) AS percentage
FROM tx_deathrow
GROUP BY county
ORDER BY percentage DESC

<u>Execution Hiatuses</u>

1) Dates
- SELECT JULIANDAY('1993-08-10') - JULIANDAY('1989-07-07') AS day_difference
- SELECT DATEDIFF(DAY, '1989-07-07', '1993-08-10') AS days_difference

2) Self joins
- SELECT ex_number + 1 AS ex_number, ex_date AS last_ex_date
FROM executions
WHERE ex_number<553
- SELECT TDCJ_number + 1 AS ex_number, Execution_Date AS last_ex_date
FROM tx_deathrow
WHERE TDCJ_number < 553

3) Nest the query which generates the previous table into the template

- SELECT last_ex_date AS start, ex_date AS end, JULIANDAY(ex_date) - JULIANDAY(last_ex_date) AS day_difference FROM executions
JOIN (SELECT ex_number + 1 AS ex_number, ex_date AS last_ex_date FROM executions) previous ON executions.ex_number = previous.ex_number
ORDER BY day_difference DESC
LIMIT 10

- SELECT TOP 10 executions.Execution_Date AS start_date, previous.Execution_Date AS end_date, DATEDIFF(DAY, previous.Execution_Date, executions.Execution_Date) AS day_difference
FROM tx_deathrow executions
JOIN (SELECT TDCJ_number + 1 AS TDCJ_number, Execution_Date FROM tx_deathrow) previous
ON executions. TDCJ_number = previous. TDCJ_number
ORDER BY day_difference DESC

4) Fill in the join clause to complete a more elegant version of the previous query

- SELECT previous.ex_date AS start, executions.ex_date AS end, JULIANDAY(executions.ex_date) - JULIANDAY(previous.ex_date) AS day_difference
FROM executions
JOIN executions previous
ON executions.ex_number = previous.ex_number + 1
ORDER BY day_difference DESC
LIMIT 10

- SELECT TOP 10 executions.Execution_Date AS start_date, previous.Execution_Date AS end_date, DATEDIFF(DAY, previous.Execution_Date, executions.Execution_Date) AS day_difference
FROM tx_deathrow executions
JOIN tx_deathrow previous
ON executions. TDCJ_number = previous. TDCJ_number + 1
ORDER BY day_difference DESC