**Lulus Coding Test**

Wade Johnson

3/22/21

1) **Wanda Young**

SELECT e.first\_name, e.last\_name, MAX(s.salary)

FROM employees e

INNER JOIN salary s ON

e.id = s.employee\_id;

2) **2009**

SELECT COUNT(first\_name),strftime('%Y', hire\_date)

FROM employees

GROUP BY strftime('%Y', hire\_date)

ORDER BY COUNT(first\_name) DESC;

3) **Feb 2005 for $372,474.08**

SELECT strftime('%Y-%m', order\_date), COUNT(order\_number)

FROM orders

GROUP BY strftime('%Y-%m', order\_date)

ORDER BY COUNT(order\_number) DESC;

--------------------------------------------------------------

SELECT SUM(op.quantity \* p.price) FROM order\_products op

INNER JOIN orders o ON op.order\_number = o.order\_number

INNER JOIN products p ON op.product\_id = p.product\_id

WHERE strftime('%Y-%m', o.order\_date) = '2005-02';

4) **5**

SELECT COUNT(order\_number) FROM orders

WHERE strftime('%Y-%m', order\_date) = '2015-07';

5) **79**

SELECT COUNT(order\_number), customer\_id

FROM orders

GROUP BY customer\_id

ORDER BY COUNT(order\_number) DESC;

6)

**project\_id avg salary ($)**

**1 129,266.32**

**2 124,610.49**

**3 129,090.07**

**4 130,816.37**

**5 120,208.61**

**6 127,734.63**

**7 127,379.25**

**8 120,474.49**

**9 131,985.7**

**10 122,070.76**

SELECT p.project\_id, ROUND(AVG(s.salary),2) FROM projects p

INNER JOIN salary s USING(employee\_id)

GROUP BY p.project\_id

7) **customer\_id 84**

SELECT

o.customer\_id,

SUM((op.quantity \* p.price) - o.shipping\_cost - IFNULL(o.discount\_amount,0))

FROM order\_products op

INNER JOIN orders o ON op.order\_number = o.order\_number

INNER JOIN products p ON op.product\_id = p.product\_id

GROUP BY o.customer\_id

ORDER BY

SUM((op.quantity \* p.price) - o.shipping\_cost - IFNULL(o.discount\_amount,0)) DESC;

8) **Kuravent-284301307-0 is the most ordered product by quantity**

SELECT p.name, SUM(op.quantity) FROM order\_products op

INNER JOIN products p USING (product\_id)

GROUP BY p.name

ORDER BY SUM(op.quantity) DESC;

**BE-873161309-8 is the least ordered product by quantity**

SELECT p.name, SUM(op.quantity) FROM order\_products op

INNER JOIN products p USING (product\_id)

GROUP BY p.name

ORDER BY SUM(op.quantity);

9) **Diana Hawkins and Ryan Ruiz have both worked on 5 separate projects**

SELECT COUNT(p.project\_id), e.first\_name, e.last\_name FROM projects p

INNER JOIN employees e ON p.employee\_id = e.id

GROUP BY e.id

ORDER BY COUNT(p.project\_id) DESC;

**Diana Hawkins has worked 467 project hours. Assuming a 2,080 hour work year, she has earned $33,448.93 on projects.**

SELECT SUM(p.project\_hours), AVG(s.salary) FROM employees e

INNER JOIN projects p ON e.id = p.employee\_id

INNER JOIN salary s ON e.id = s.employee\_id

WHERE e.id = 799

GROUP BY e.id;

10) **$80,497.46** (Arthur Bailey has worked 1 hour on 1 project)

SELECT COUNT(p.project\_id), e.first\_name, e.last\_name, p.project\_hours, s.salary FROM projects p

INNER JOIN employees e ON p.employee\_id = e.id

INNER JOIN salary s ON s.employee\_id = e.id

GROUP BY e.id

ORDER BY COUNT(p.project\_id), project\_hours;

11) **Category 4 @ $5,182,085.38**

SELECT c.category\_name, SUM(p.price \* op.quantity) FROM products p

INNER JOIN categories c USING (category\_id)

INNER JOIN order\_products op USING (product\_id)

GROUP BY category\_name

ORDER BY SUM(p.price \* op.quantity) DESC;

12) **Returns a table with 183 rows representing each month. I struggled to combine the two queries into a single table.**

SELECT op.quantity, strftime('%Y-%m', o.order\_date)

FROM orders o

INNER JOIN order\_products op USING (order\_number)

GROUP BY strftime('%Y-%m', o.order\_date)

HAVING MIN(o.order\_date)

-----------------------------------------------

SELECT op.quantity, strftime('%Y-%m', o.order\_date)

FROM orders o

INNER JOIN order\_products op USING (order\_number)

GROUP BY strftime('%Y-%m', o.order\_date)

HAVING MAX(o.order\_date)

13) **I used SQLite in order to get up and running quickly.**

14) **Workflow:** Downloaded the CSVs --> Imported them into SQLite using DB Browser