David Willis

Lingma Acheson

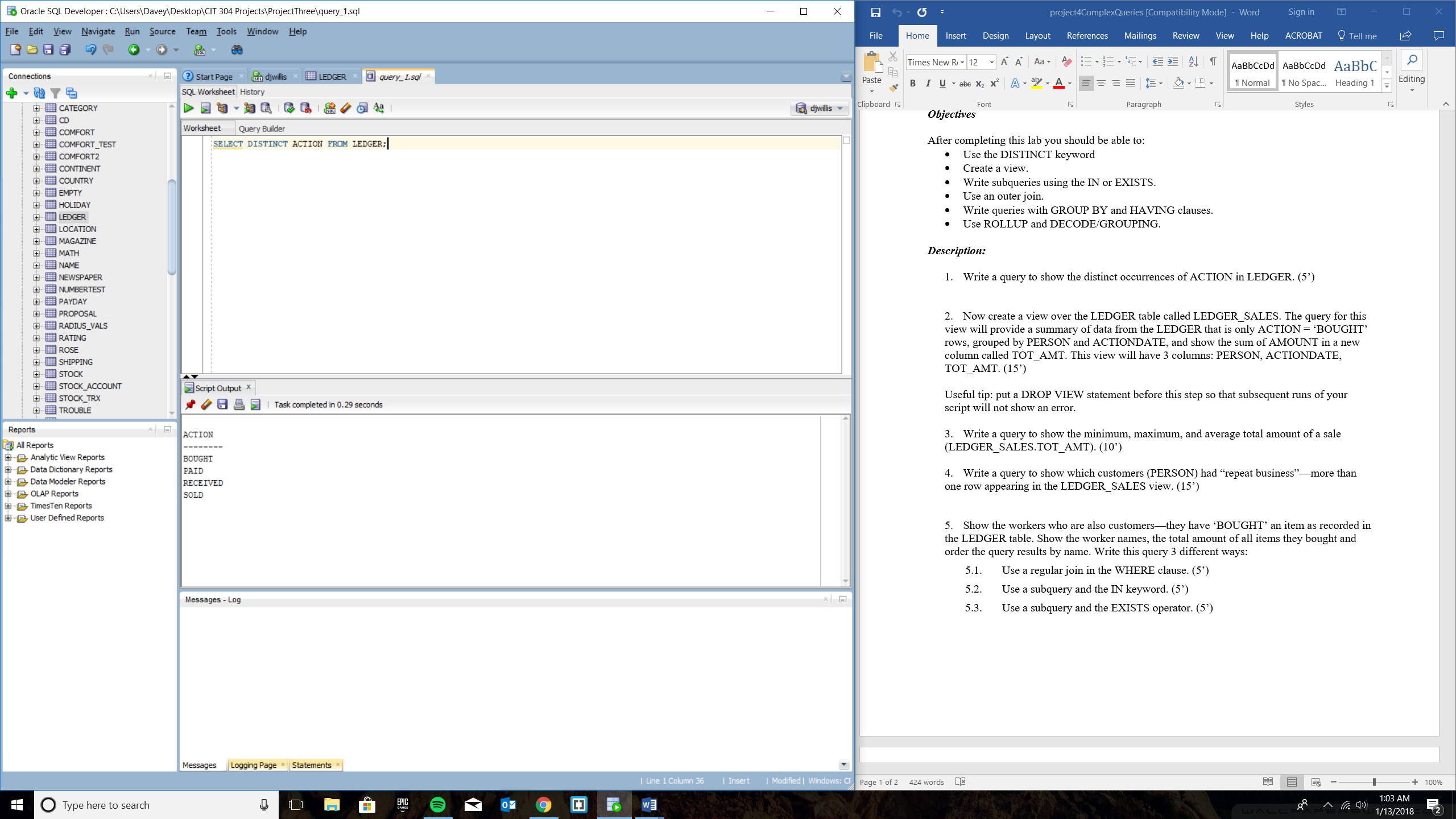
CIT 304

2.6.18

Course Project 4

Query 1 – Selects distinct occurrences of values from the ACTION column of the LEDGER table, meaning that there will be no repeating values, only showing the possible values from the totality of the ACTION column

SELECT DISTINCT ACTION FROM LEDGER;



Query 2 – Creates a view that shows a table of only the rows of whom an organization or person(s) purchased items. The row will only contain the date they bought the item, the person(s) or organization themselves, and a newly added column called “TOT\_AMT” that contains the total amount of the whole purchase (by taking the quantity times the price).

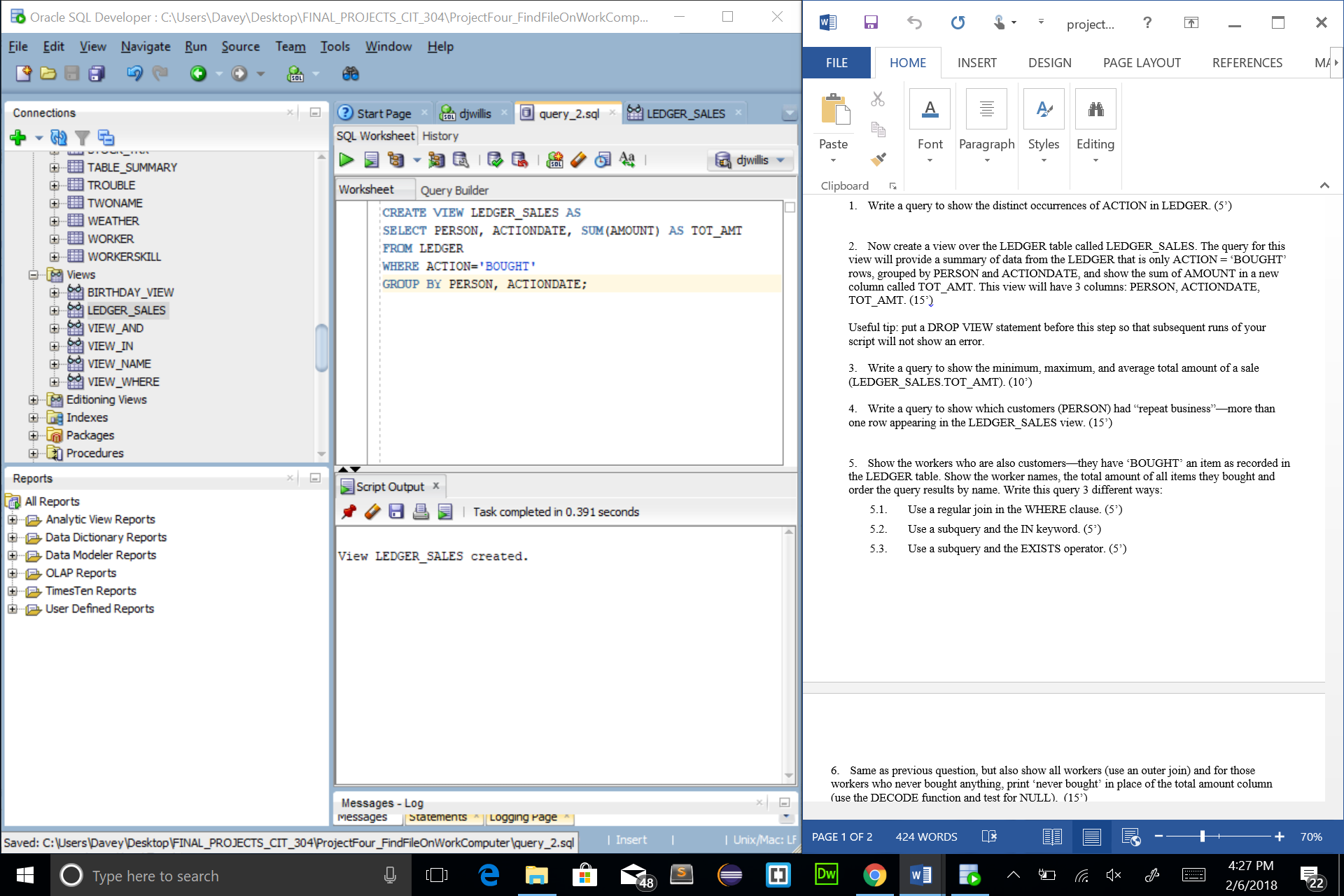
CREATE VIEW LEDGER\_SALES AS

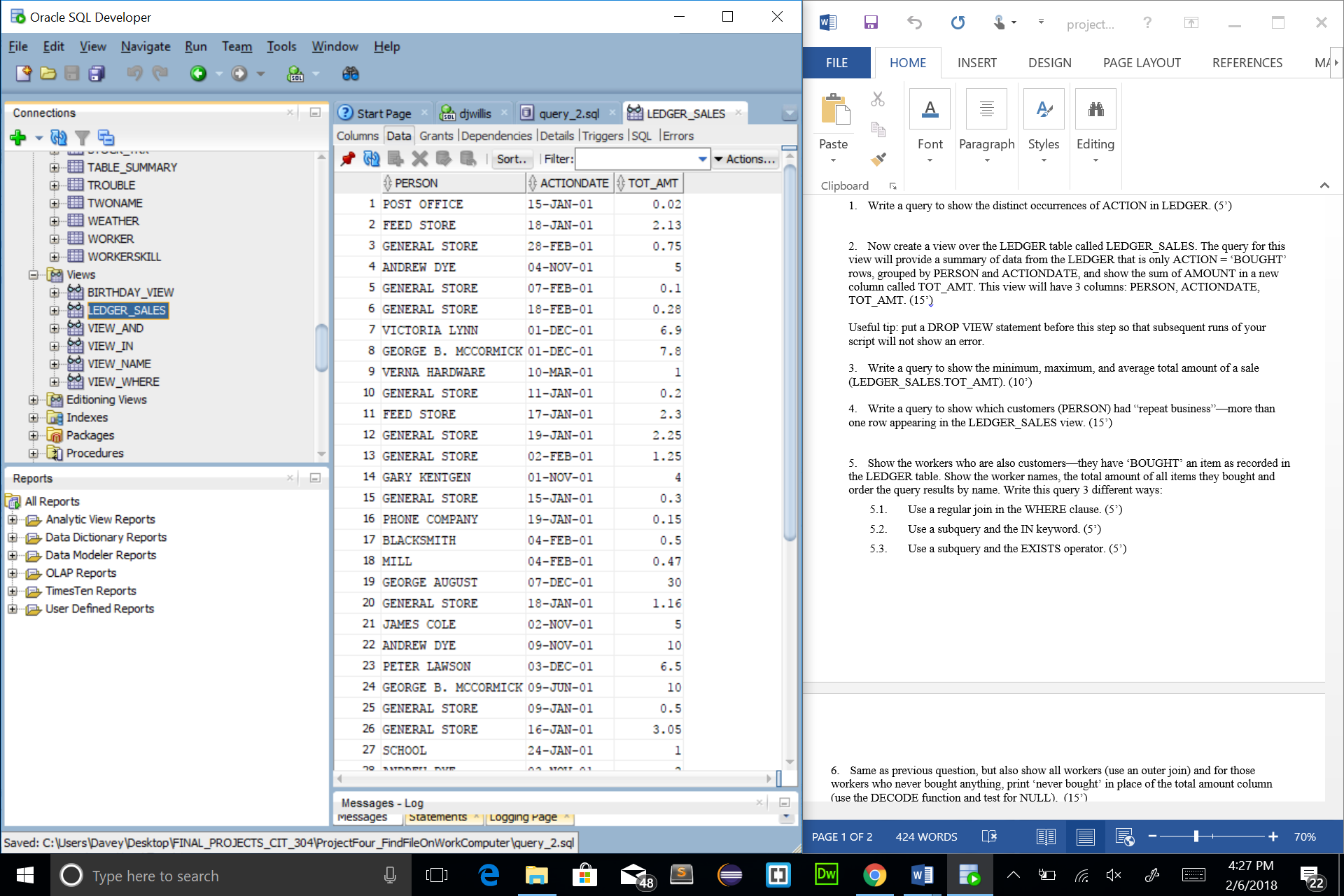
SELECT PERSON, ACTIONDATE, SUM(AMOUNT) AS TOT\_AMT

FROM LEDGER

WHERE ACTION='BOUGHT'

GROUP BY PERSON, ACTIONDATE;





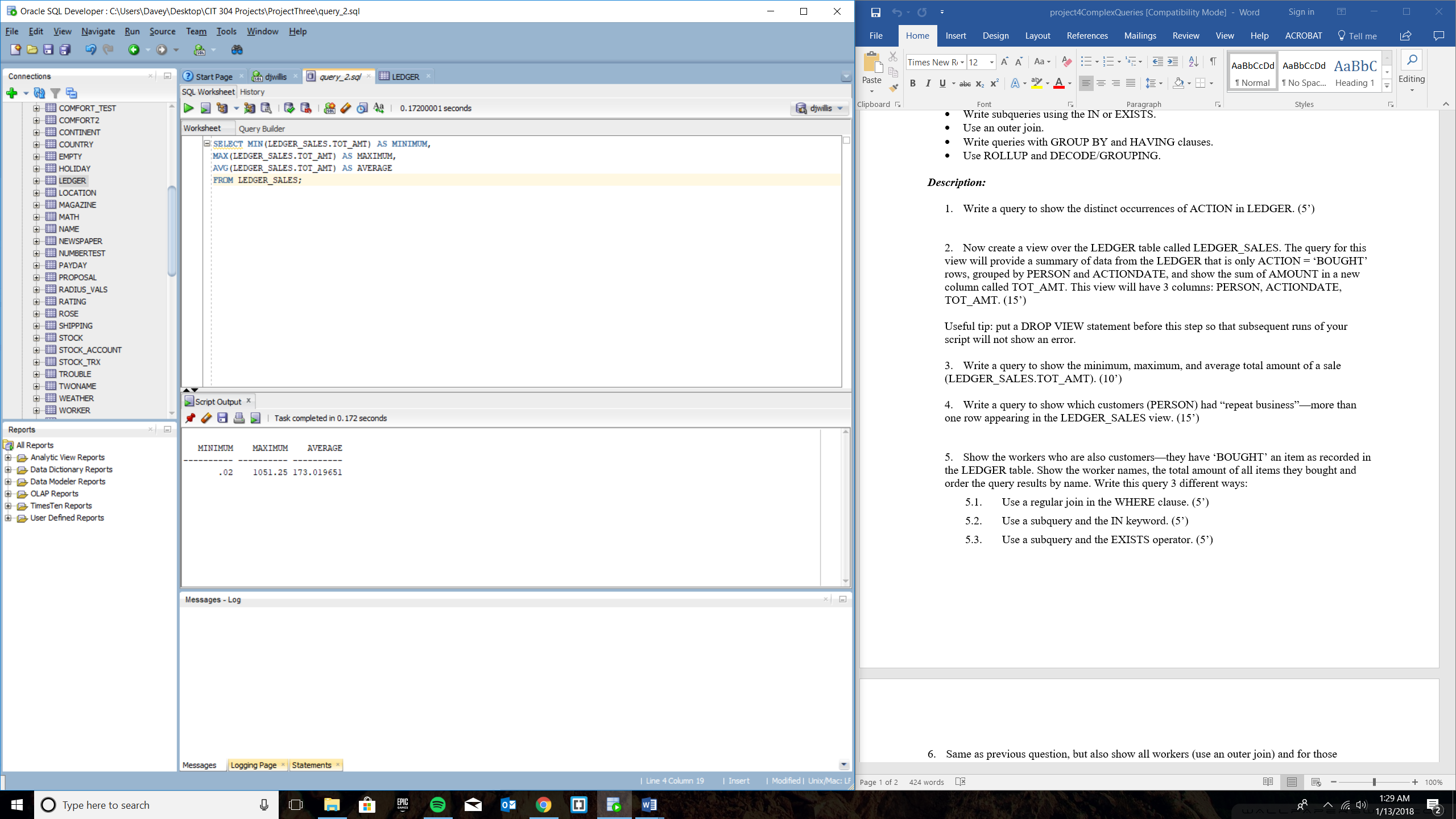
Query 3 – Selects the lowest, highest, and average of the “TOT\_AMT” column created in the last query, essentially giving the least, greatest, and average sale from the totality of the TOT\_AMT list.

SELECT MIN(LEDGER\_SALES.TOT\_AMT) AS MINIMUM,

MAX(LEDGER\_SALES.TOT\_AMT) AS MAXIMUM,

AVG(LEDGER\_SALES.TOT\_AMT) AS AVERAGE

FROM LEDGER\_SALES;



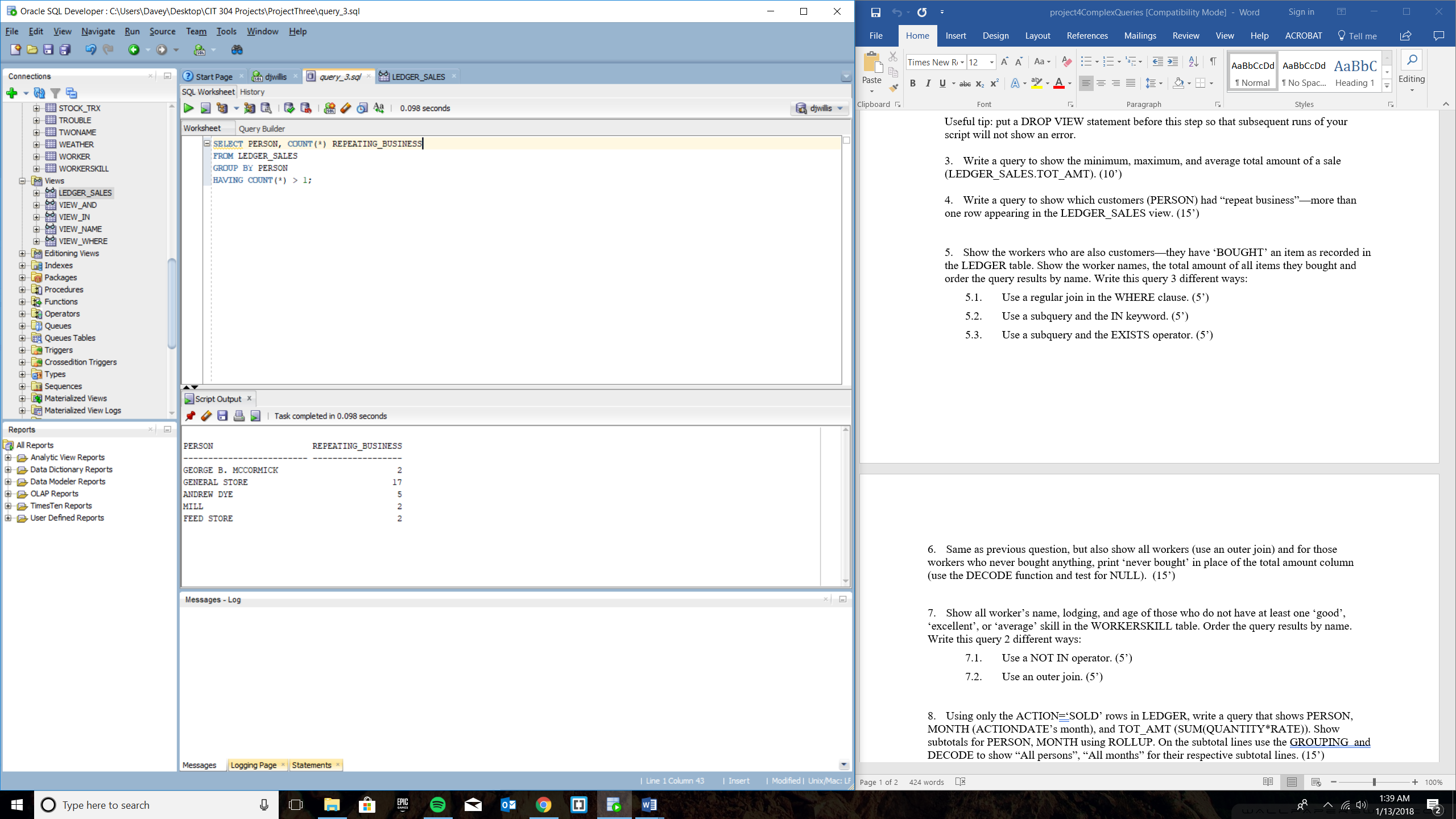
Query 4 – Creates a table showing the list of repeating customers and how many times they have returned for the company’s business

SELECT PERSON, COUNT(\*) REPEATING\_BUSINESS

FROM LEDGER\_SALES

GROUP BY PERSON

HAVING COUNT(\*) > 1;



Query 5 – A

SELECT PERSON, SUM(AMOUNT) AS TOTAL,

COUNT(PERSON) AS COUNT

FROM LEDGER

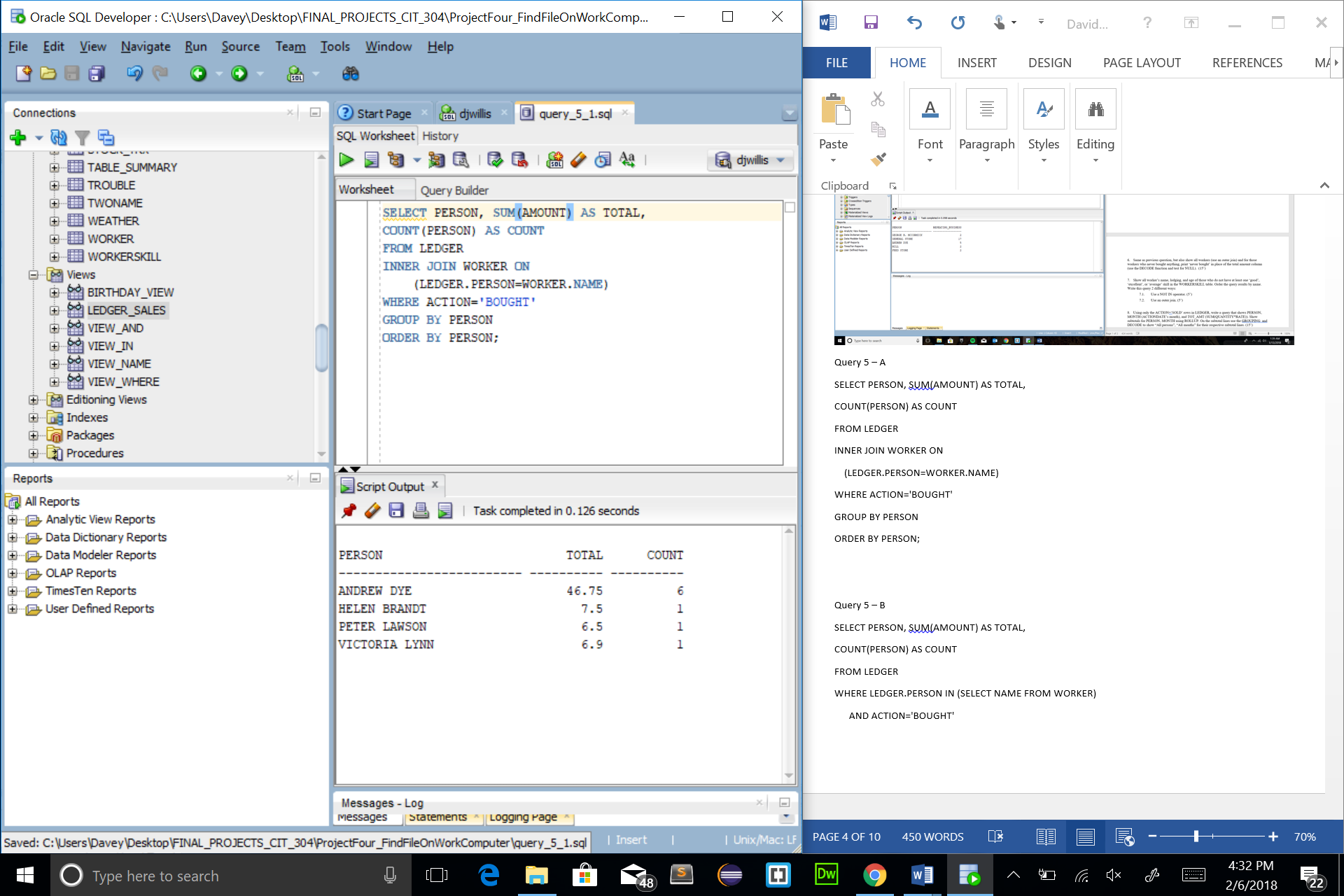
INNER JOIN WORKER ON

(LEDGER.PERSON=WORKER.NAME)

WHERE ACTION='BOUGHT'

GROUP BY PERSON

ORDER BY PERSON;



Query 5 – B

SELECT PERSON, SUM(AMOUNT) AS TOTAL,

COUNT(PERSON) AS COUNT

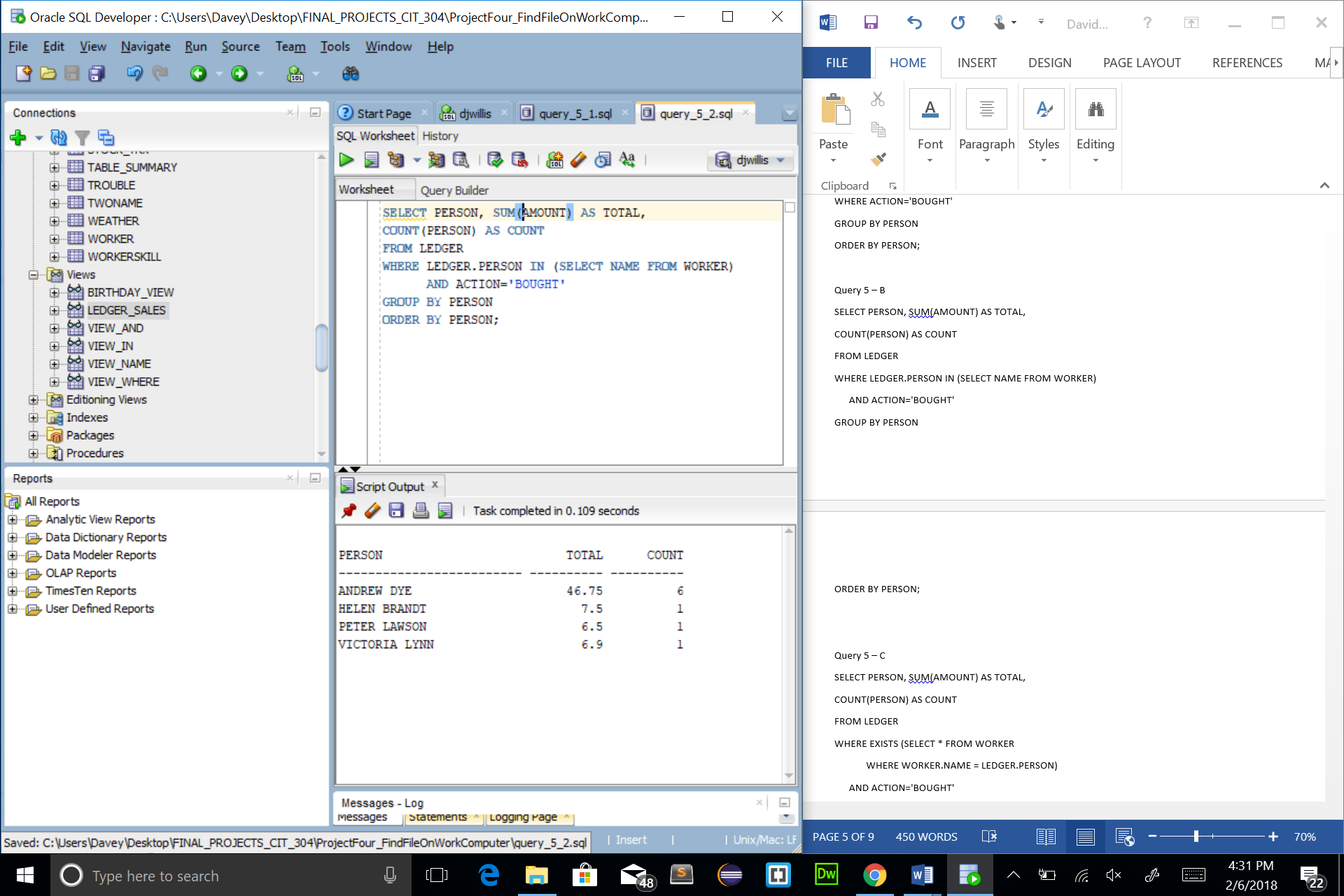
FROM LEDGER

WHERE LEDGER.PERSON IN (SELECT NAME FROM WORKER)

AND ACTION='BOUGHT'

GROUP BY PERSON

ORDER BY PERSON;



Query 5 – C

SELECT PERSON, SUM(AMOUNT) AS TOTAL,

COUNT(PERSON) AS COUNT

FROM LEDGER

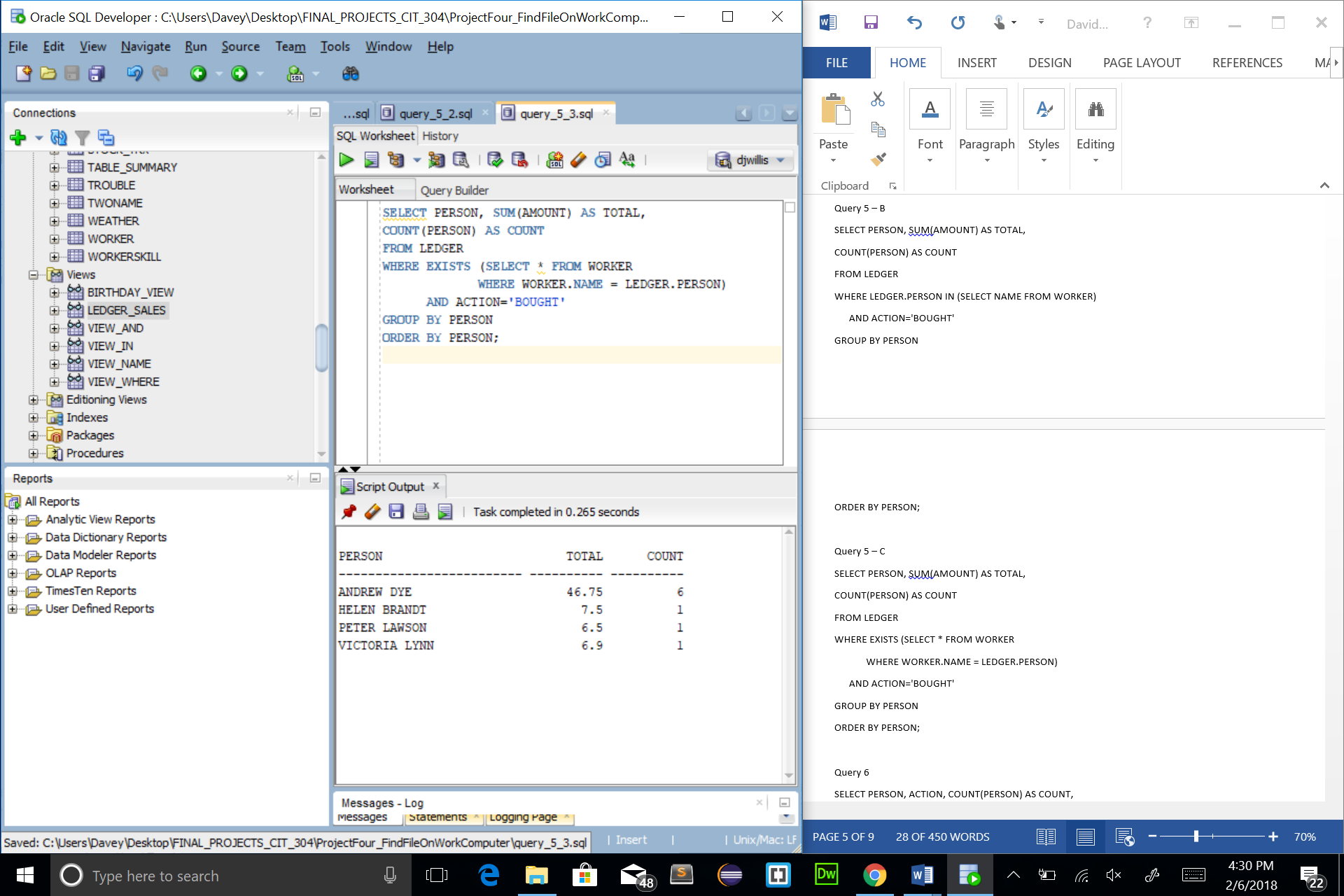
WHERE EXISTS (SELECT \* FROM WORKER

WHERE WORKER.NAME = LEDGER.PERSON)

AND ACTION='BOUGHT'

GROUP BY PERSON

ORDER BY PERSON;



Query 6

SELECT PERSON, ACTION, COUNT(PERSON) AS COUNT,

CASE ACTION

WHEN 'BOUGHT' THEN TO\_CHAR(SUM(AMOUNT))

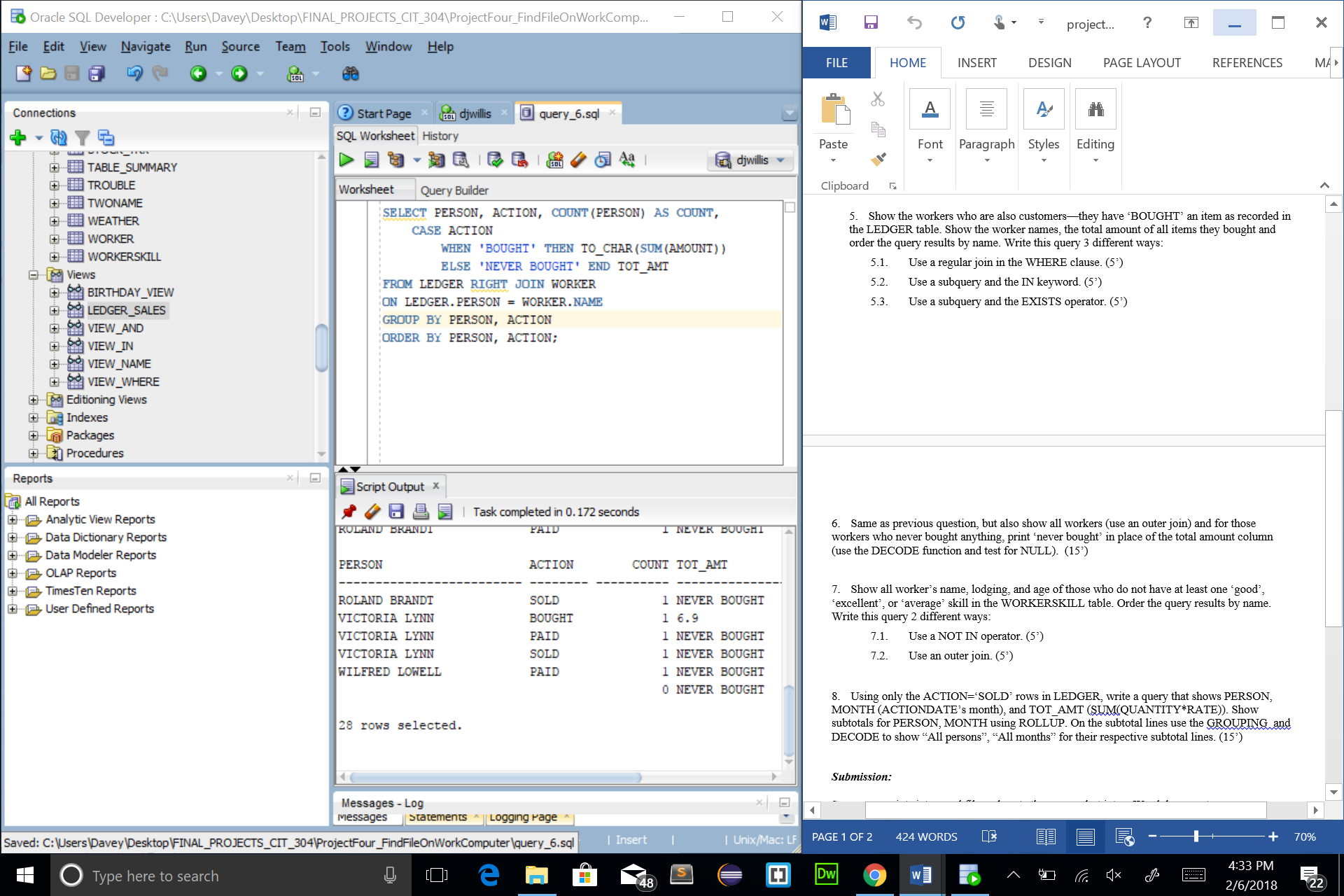
ELSE 'NEVER BOUGHT' END TOT\_AMT

FROM LEDGER RIGHT JOIN WORKER

ON LEDGER.PERSON = WORKER.NAME

GROUP BY PERSON, ACTION

ORDER BY PERSON, ACTION;



Query 7.1

SELECT WORKER.NAME, WORKER.AGE, WORKER.LODGING, SKILL, ABILITY

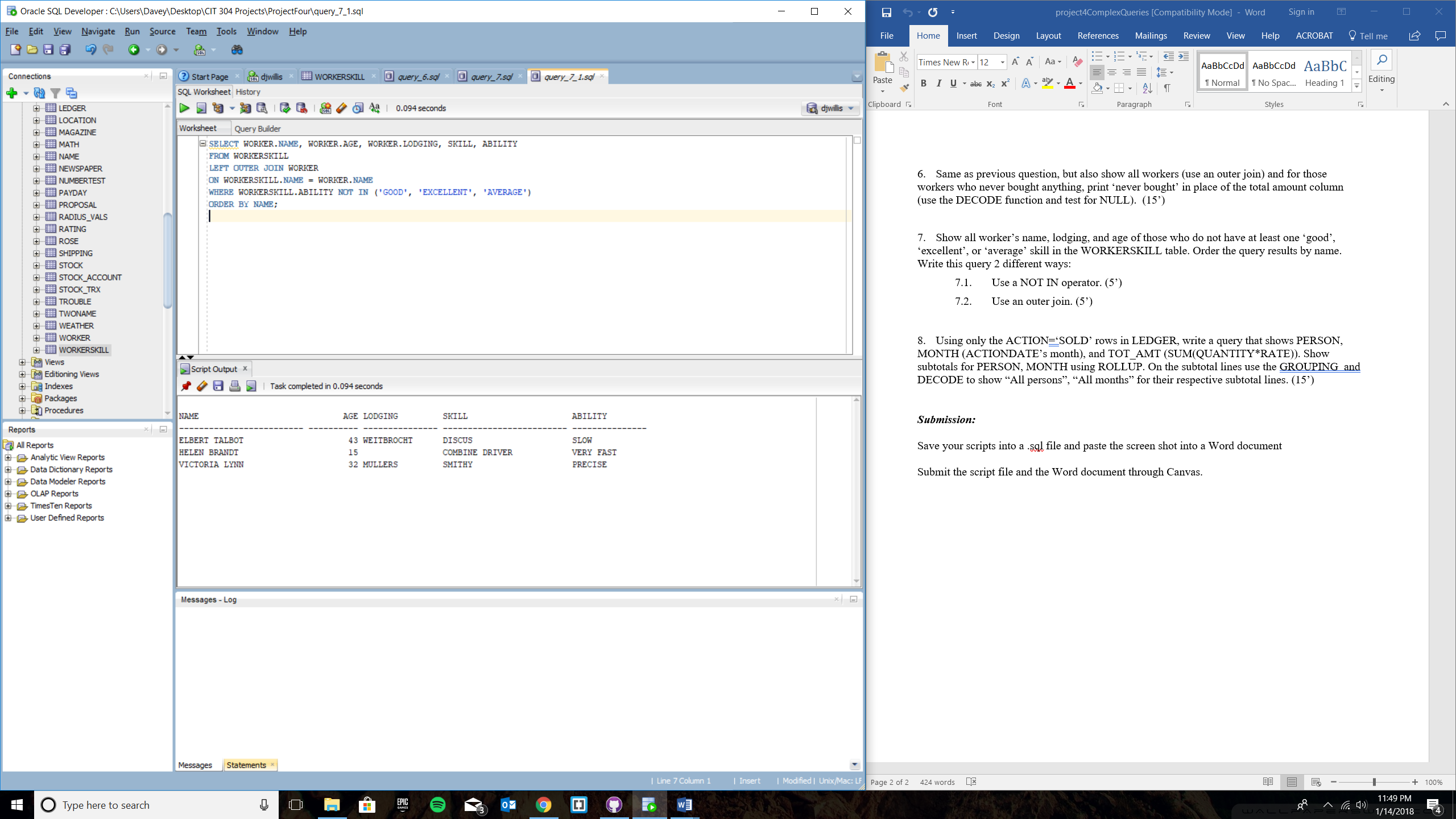
FROM WORKERSKILL

LEFT OUTER JOIN WORKER

ON WORKERSKILL.NAME = WORKER.NAME

WHERE WORKERSKILL.ABILITY NOT IN ('GOOD', 'EXCELLENT', 'AVERAGE')

ORDER BY NAME;



Query 7.2

SELECT WORKER.NAME, WORKER.AGE, WORKER.LODGING, SKILL, ABILITY

FROM WORKERSKILL

LEFT OUTER JOIN WORKER

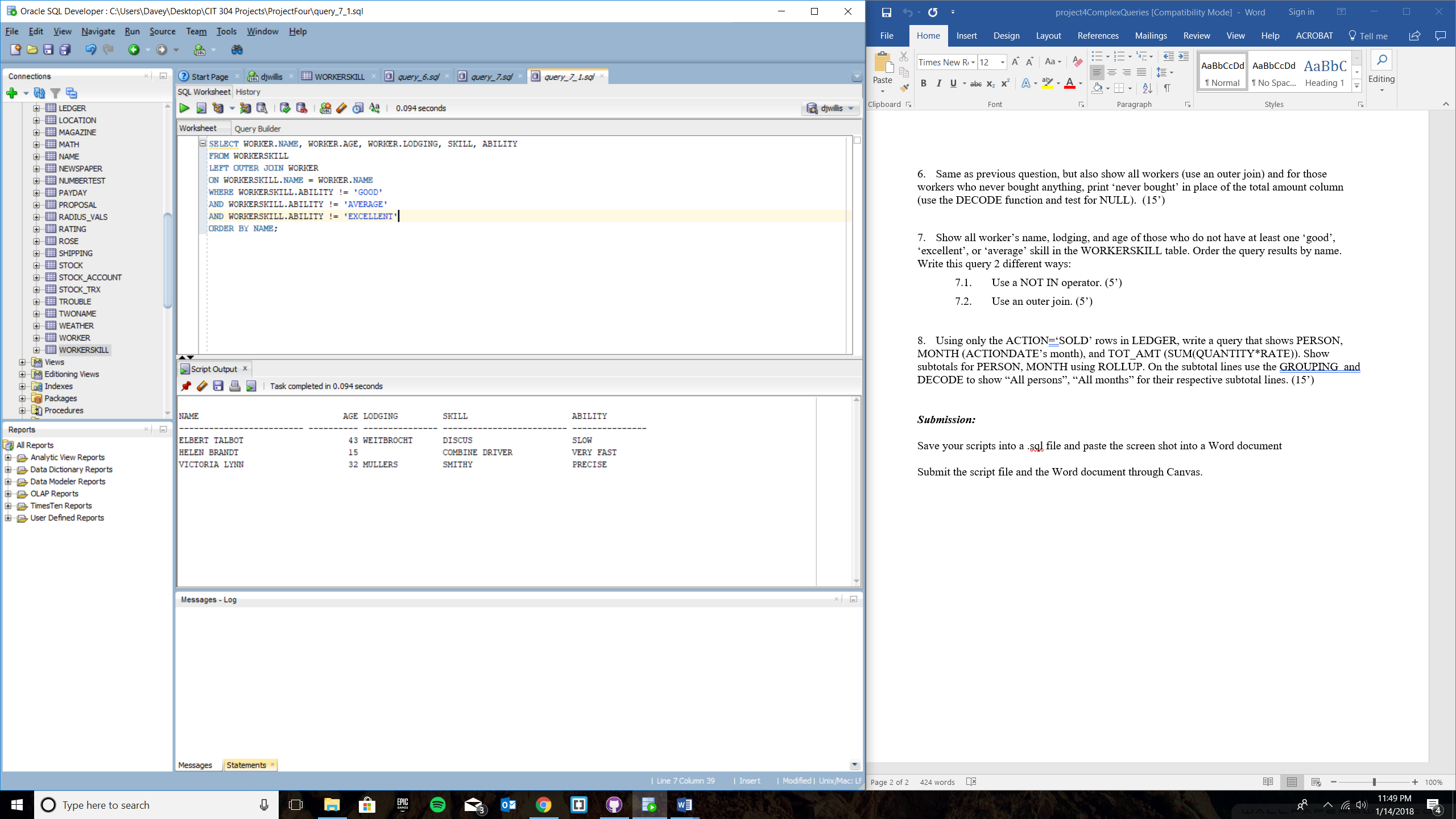
ON WORKERSKILL.NAME = WORKER.NAME

WHERE WORKERSKILL.ABILITY != 'GOOD'

AND WORKERSKILL.ABILITY != 'AVERAGE'

AND WORKERSKILL.ABILITY != 'EXCELLENT'

ORDER BY NAME;



Query 8

SELECT DECODE(

GROUPING(PERSON), 1, 'Total', PERSON) AS SUBTOTAL,

TO\_CHAR(ACTIONDATE, 'Month') AS MONTH, SUM(QUANTITY\*RATE) AS TOT\_AMT

FROM LEDGER

WHERE ACTION = 'SOLD'

GROUP BY ROLLUP (PERSON, ACTIONDATE)

ORDER BY PERSON; 