# Queries to generate output as per exhibits

## Exhibit B

Exhibit B is Wilco’s EEOC Compliance Statement form. This form has detailed work hour break down part D which has details about number of hour minority, female and non-minority employees have worked against each job classification. Total number of hours each job is performed and percentage of minority and females against total job hours of each job.

This detailed work break down is achieved using following code:

--QUERY FOR DETAIL HOURS

WITH CTE\_DETAIL\_HOURS ([NON/MINORITY], JOB\_CLASSIFICATION, GENDER, HOURS,JOB\_ID)

AS

(

SELECT

CASE WHEN E.EEO\_CODE = 5 THEN 'NON-MINORITY' ELSE 'MINORITY' END AS [NON/MINORITY],

CONCAT (RTRIM(JOB\_CODE),'- ',RTRIM(JOB\_DESCRIPTION)) AS JOB\_CLASSIFICATION,

GENDER, SUM (REGULAR\_HOURS+ OVERTIME\_HOURS) HOURS, JC.JOB\_ID

FROM EMPLOYEE\_TIME\_CARD ETC

JOIN JOB\_CLASSIFICATION JC ON ETC.JOB\_ID = JC.JOB\_ID

JOIN EMPLOYEE E ON ETC.EMP\_ID = E.EMP\_ID

JOIN PROJECT P ON JC.PROJECT\_ID = P.PROJECT\_ID

WHERE PROJECT\_NUMBER = 'OH-PIK-335-005'

GROUP BY

CASE WHEN E.EEO\_CODE = 5 THEN 'NON-MINORITY' ELSE 'MINORITY' END,

JOB\_CODE,

JOB\_DESCRIPTION,

GENDER,

JC.JOB\_ID),

-- QUERY FOR TOTAL HOURS

CTE\_TOTALHOURS (JOB\_ID, TOTAL\_HOURS)

AS

(

SELECT JC.JOB\_ID, SUM(REGULAR\_HOURS+ OVERTIME\_HOURS) TOTAL\_HOURS

FROM EMPLOYEE\_TIME\_CARD ETC

JOIN JOB\_CLASSIFICATION JC ON ETC.JOB\_ID = JC.JOB\_ID

JOIN PROJECT P ON JC.PROJECT\_ID = P.PROJECT\_ID

WHERE PROJECT\_NUMBER = 'OH-PIK-335-005'

GROUP BY

JC.JOB\_ID)

-- QUERY TO CALCULATE TOTAL HOURS WORKED BY MINORITY

,CTE\_MINORITY\_HOURS (JOB\_ID,MINORITY\_HOURS)

AS

(

SELECT JC.JOB\_ID, SUM (REGULAR\_HOURS+ OVERTIME\_HOURS) TOTAL\_HOURS

FROM EMPLOYEE\_TIME\_CARD ETC

JOIN JOB\_CLASSIFICATION JC ON ETC.JOB\_ID = JC.JOB\_ID

JOIN PROJECT P ON JC.PROJECT\_ID = P.PROJECT\_ID

JOIN EMPLOYEE E ON ETC.EMP\_ID = E.EMP\_ID

WHERE PROJECT\_NUMBER = 'OH-PIK-335-005'

AND EEO\_CODE != 5

GROUP BY

JC.JOB\_ID

)

--- MAIN QUERY

SELECT

JOB\_CLASSIFICATION,

[NON/MINORITY],

CASE WHEN GENDER = 'M' THEN 'MALE'

WHEN GENDER = 'F' THEN 'FEMALE' END AS GENDER,

HOURS,

TOTAL\_HOURS,

CASE WHEN [NON/MINORITY] = 'MINORITY' THEN (SUM(MINORITY\_HOURS)/TOTAL\_HOURS)\*100

END AS MINORITYPERCENT,

CASE WHEN GENDER = 'F' THEN (SUM(HOURS)/TOTAL\_HOURS)\*100 END AS FEMALEPERSENT

FROM CTE\_DETAIL\_HOURS CDH

JOIN CTE\_TOTALHOURS CTH ON CDH.JOB\_ID = CTH.JOB\_ID

JOIN CTE\_MINORITY\_HOURS CMH ON CDH.JOB\_ID = CMH.JOB\_ID

GROUP BY

[NON/MINORITY],

JOB\_CLASSIFICATION,

GENDER,

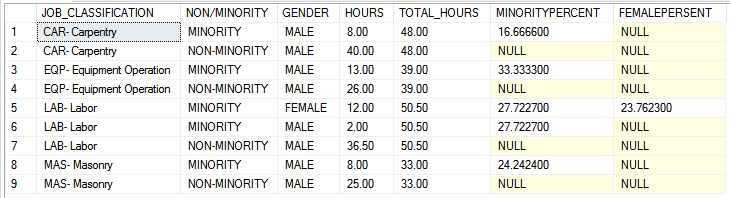
HOURS,

TOTAL\_HOURS

ORDER BY JOB\_CLASSIFICATION

The CTE\_DETAIL\_HOURS Common table expression (CTE) will give me number of hour worked by male and female grouped by minority and non-minority against each job classification. The CTE\_TOTAL\_HOURS calculates total number of hour worked by all employees against each job classification. The CTE\_MINORITY\_HOURS calculates number of hour worked by minority males and females against each job classification. The final select statement produces below output. It calculating the percentage of job hours worked by minority and female and displays field from above mentioned CTE. The CTEs are joined on the basis of Job Id present in each CTE.

Output:



## Exhibit D

Exhibit D shows the number of regular, overtime and total hours worked by each employee in a particular project. It also shows the skill details, rate per skill and gross payment done to each employee.

Part D- Regular hours query

-- PART D

SELECT E.SOCIAL\_SECURITY\_NUMBER AS [EMPLOYEE NUMBER], JC.JOB\_CODE AS [SKILL CODE],

JC.BASIC\_HOURLY\_RATE AS RATE, JC.FRINGE\_BENEFITS AS FRINGE,

(JC.BASIC\_HOURLY\_RATE + JC.FRINGE\_BENEFITS) AS TOTAL,

SUM(ET.REGULAR\_HOURS) AS HOURS,

((JC.BASIC\_HOURLY\_RATE + JC.FRINGE\_BENEFITS)\* ET.REGULAR\_HOURS) AS GROSS,PROJECT\_NUMBER

FROM

EMPLOYEE E, JOB\_CLASSIFICATION JC, EMPLOYEE\_TIME\_CARD ET, PROJECT P

WHERE E.EMP\_ID = ET.EMP\_ID

AND ET.JOB\_ID = JC.JOB\_ID

AND JC.PROJECT\_ID = P.PROJECT\_ID

AND REGULAR\_HOURS != 0

AND DATE BETWEEN '05-10-1995' AND '05-16-1995'

GROUP BY

E.SOCIAL\_SECURITY\_NUMBER , JC.JOB\_CODE ,

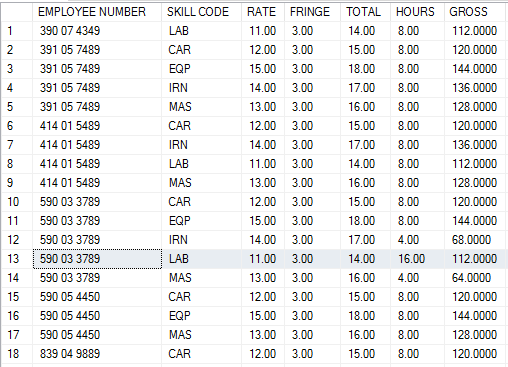
JC.BASIC\_HOURLY\_RATE, JC.FRINGE\_BENEFITS ,

(JC.BASIC\_HOURLY\_RATE + JC.FRINGE\_BENEFITS), PROJECT\_NUMBER,

((JC.BASIC\_HOURLY\_RATE + JC.FRINGE\_BENEFITS)\* ET.REGULAR\_HOURS)

In this query regular hours worked by each employee on each job is displayed along with the basic hourly rate and gross amount i.e. number of hours\* basic hourly rate. Date filter is used as this report is based on work period. In this case it will show all the details from date 05/10/1995 to 05/16/1995.

Output:



Output shows list of all employees and number of regular hours they have worked against each skill on a particular project. It also shows the basic hourly rate, fringe rate, total rate i.e. sum of the basic hourly rate and fringe rate. Finally it shows the gross amount.

Part E- Overtime hours query

-- PART E

SELECT E.SOCIAL\_SECURITY\_NUMBER AS [EMPLOYEE NUMBER],JC.JOB\_CODE AS [SKILL CODE],

JC.OVERTIME\_RATE AS RATE, JC.FRINGE\_BENEFITS AS FRINGE,

(JC.OVERTIME\_RATE + JC.FRINGE\_BENEFITS) AS TOTAL,

SUM(ET.OVERTIME\_HOURS) AS HOURS,

((JC.OVERTIME\_RATE + JC.FRINGE\_BENEFITS)\* ET.OVERTIME\_HOURS) AS GROSS, PROJECT\_NUMBER

FROM

EMPLOYEE E, JOB\_CLASSIFICATION JC, EMPLOYEE\_TIME\_CARD ET, PROJECT P

WHERE E.EMP\_ID = ET.EMP\_ID

AND ET.JOB\_ID = JC.JOB\_ID

AND JC.PROJECT\_ID = P.PROJECT\_ID

AND OVERTIME\_HOURS != 0

AND DATE BETWEEN '05-10-1995' AND '05-16-1995'

GROUP BY

E.SOCIAL\_SECURITY\_NUMBER,JC.JOB\_CODE ,

JC.OVERTIME\_RATE,JC.FRINGE\_BENEFITS ,

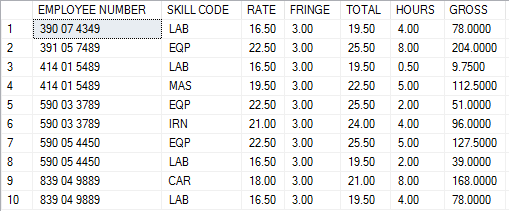
(JC.OVERTIME\_RATE + JC.FRINGE\_BENEFITS),

((JC.OVERTIME\_RATE + JC.FRINGE\_BENEFITS)\* ET.OVERTIME\_HOURS) ,

PROJECT\_NUMBER

In this query overtime hours worked by each employee on each job is displayed. It also displays overtime hourly rate and gross amount i.e. number of hours\* overtime rate. Date filter is used as this report is based on work period. In this case it will show the details from date 05/10/1995 to 05/16/1995.

Output:



Output shows list of all employees who worked overtime against each skill and the overtime rate for a particular project. It also shows the basic hourly rate, fringe rate, total rate i.e. sum of the basic hourly rate and fringe rate. Finally it shows the gross amount.

Part F- Total hours worked per skill classification.

--PART F

SELECT JOB\_CODE, SUM(REGULAR\_HOURS) REGULAR\_HOURS,

SUM(OVERTIME\_HOURS) OVERTIME\_HOURS,

SUM(REGULAR\_HOURS+OVERTIME\_HOURS) TOTAL\_HOURS

FROM

JOB\_CLASSIFICATION JC, EMPLOYEE\_TIME\_CARD ET, PROJECT P

WHERE

ET.JOB\_ID = JC.JOB\_ID

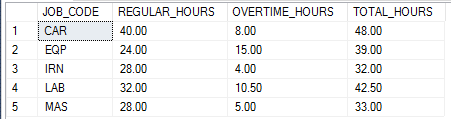
AND JC.PROJECT\_ID = P.PROJECT\_ID

AND P.PROJECT\_NUMBER = 'OH-PIK-335-005'

GROUP BY JOB\_CODE

This query shows job code and calculates regular hours, overtime hours and total number of hours worked by all employees on a given project.

Output:



## Exhibit F

Exhibit F calculates the payroll of each employee. In this section we are calculating regular pay, overtime pay paid to each employee for a given week and the gross amount.

Regular Pay query:

SELECT PROJECT\_NUMBER AS JOB, JC.JOB\_CODE AS [SKILL CODE],

JC.BASIC\_HOURLY\_RATE AS RATE, JC.FRINGE\_BENEFITS AS FRINGE,

(JC.BASIC\_HOURLY\_RATE + JC.FRINGE\_BENEFITS) AS TOTAL,

SUM(ET.REGULAR\_HOURS) AS HOURS,

((JC.BASIC\_HOURLY\_RATE + JC.FRINGE\_BENEFITS)\* ET.REGULAR\_HOURS) AS GROSS

FROM

EMPLOYEE E, JOB\_CLASSIFICATION JC, EMPLOYEE\_TIME\_CARD ET, PROJECT P

WHERE E.EMP\_ID = ET.EMP\_ID

AND ET.JOB\_ID = JC.JOB\_ID

AND JC.PROJECT\_ID = P.PROJECT\_ID

AND REGULAR\_HOURS != 0

AND DATE BETWEEN '05-10-1995' AND '05-16-1995'

AND SOCIAL\_SECURITY\_NUMBER = '391 05 7489'

GROUP BY

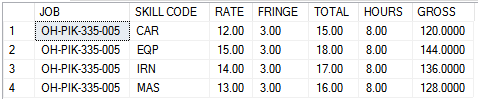
E.SOCIAL\_SECURITY\_NUMBER , JC.JOB\_CODE ,

JC.BASIC\_HOURLY\_RATE, JC.FRINGE\_BENEFITS ,

(JC.BASIC\_HOURLY\_RATE + JC.FRINGE\_BENEFITS), PROJECT\_NUMBER,

((JC.BASIC\_HOURLY\_RATE + JC.FRINGE\_BENEFITS)\* ET.REGULAR\_HOURS)

Output:



This query gives the list of all project and skills an employee has worked in a given week. It calculates total rate per hour, total number of hours and gross pay.

Overtime Pay query:

SELECT PROJECT\_NUMBER as JOB,JC.JOB\_CODE AS [SKILL CODE],

JC.OVERTIME\_RATE AS RATE, JC.FRINGE\_BENEFITS AS FRINGE,

(JC.OVERTIME\_RATE + JC.FRINGE\_BENEFITS) AS TOTAL,

SUM(ET.OVERTIME\_HOURS) AS HOURS,

((JC.OVERTIME\_RATE + JC.FRINGE\_BENEFITS)\* ET.OVERTIME\_HOURS) AS GROSS

FROM

EMPLOYEE E, JOB\_CLASSIFICATION JC, EMPLOYEE\_TIME\_CARD ET, PROJECT P

WHERE E.EMP\_ID = ET.EMP\_ID

AND ET.JOB\_ID = JC.JOB\_ID

AND JC.PROJECT\_ID = P.PROJECT\_ID

AND OVERTIME\_HOURS != 0

AND DATE BETWEEN '05-10-1995' AND '05-16-1995'

AND SOCIAL\_SECURITY\_NUMBER = '391 05 7489'

GROUP BY

E.SOCIAL\_SECURITY\_NUMBER,JC.JOB\_CODE ,

JC.OVERTIME\_RATE,JC.FRINGE\_BENEFITS ,

(JC.OVERTIME\_RATE + JC.FRINGE\_BENEFITS),

((JC.OVERTIME\_RATE + JC.FRINGE\_BENEFITS)\* ET.OVERTIME\_HOURS) ,

PROJECT\_NUMBER

Output:



This query gives the list of all project and skills an employee has worked overtime in a given week. It calculates total overtime rate per hour, total number of overtime hours and gross pay.

# Conclusion

The modified database design proposed in this report addresses EEOC compliance pressing problem. The implementation of the above mentioned code in exhibit B generates EEOC compliance report data. This data can be further used in a reporting tool to generate the final report. This database design also helps to simplify payroll calculation which is being done manually by Mary. This code shown in exhibit E helps implementation of payroll. Another important aspect being covered by the database design is the tracking of inventory. Overall this database design addresses most of the problems faced by Wilco Construction Company.