Case Study: City Jail

1. List all criminals along with the crime charges filed. The report needs to include the criminal

ID, name, crime code, and fine amount.

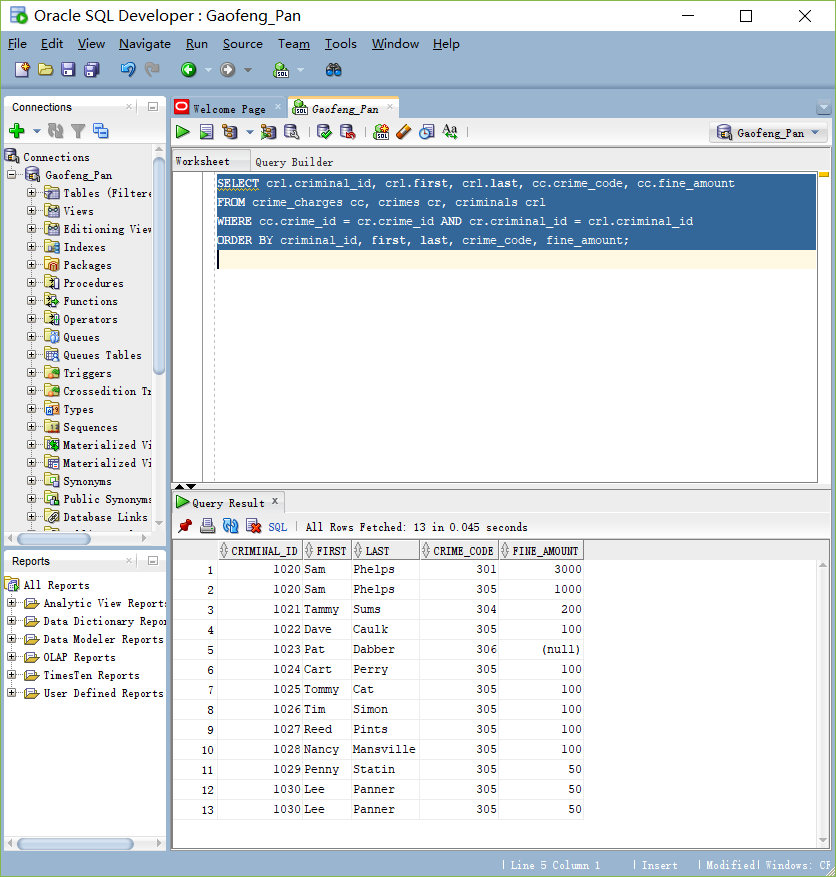
Traditional Method

SELECT crl.criminal\_id, crl.first, crl.last, cc.crime\_code, cc.fine\_amount

FROM crime\_charges cc, crimes cr, criminals crl

WHERE cc.crime\_id = cr.crime\_id AND cr.criminal\_id = crl.criminal\_id

ORDER BY criminal\_id, first, last, crime\_code, fine\_amount;



Join Method

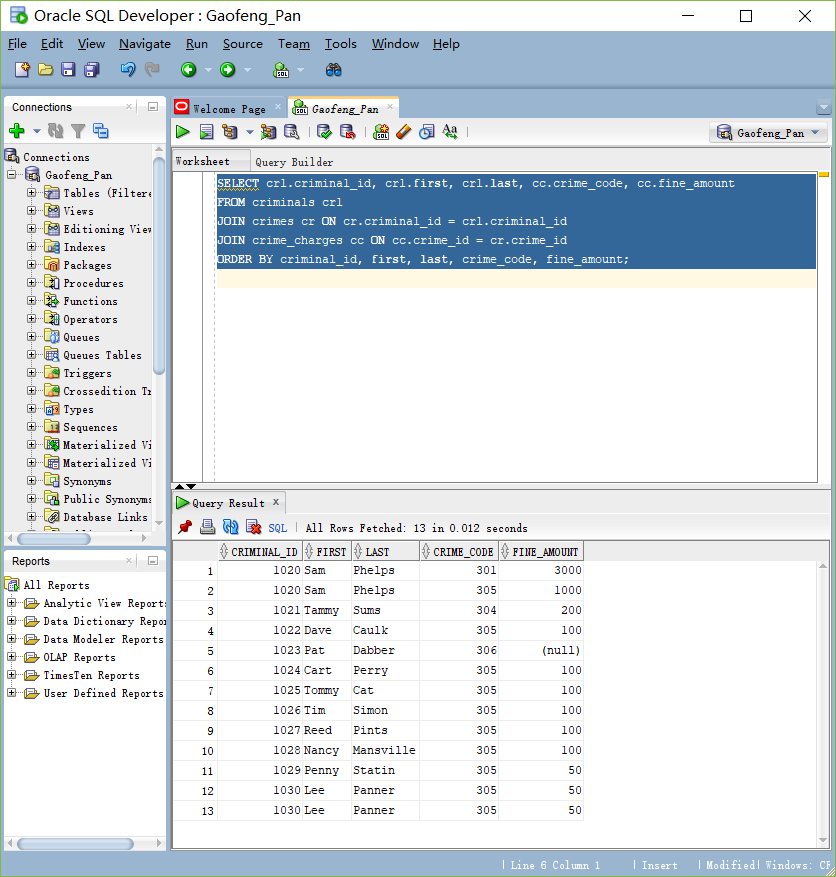
SELECT crl.criminal\_id, crl.first, crl.last, cc.crime\_code, cc.fine\_amount

FROM criminals crl

JOIN crimes cr ON cr.criminal\_id = crl.criminal\_id

JOIN crime\_charges cc ON cc.crime\_id = cr.crime\_id

ORDER BY criminal\_id, first, last, crime\_code, fine\_amount;



2. List all criminals along with crime status and appeal status (if applicable). The reports need

to include the criminal ID, name, crime classification, date charged, appeal filing date, and

appeal status. Show all criminals, regardless of whether they have filed an appeal.

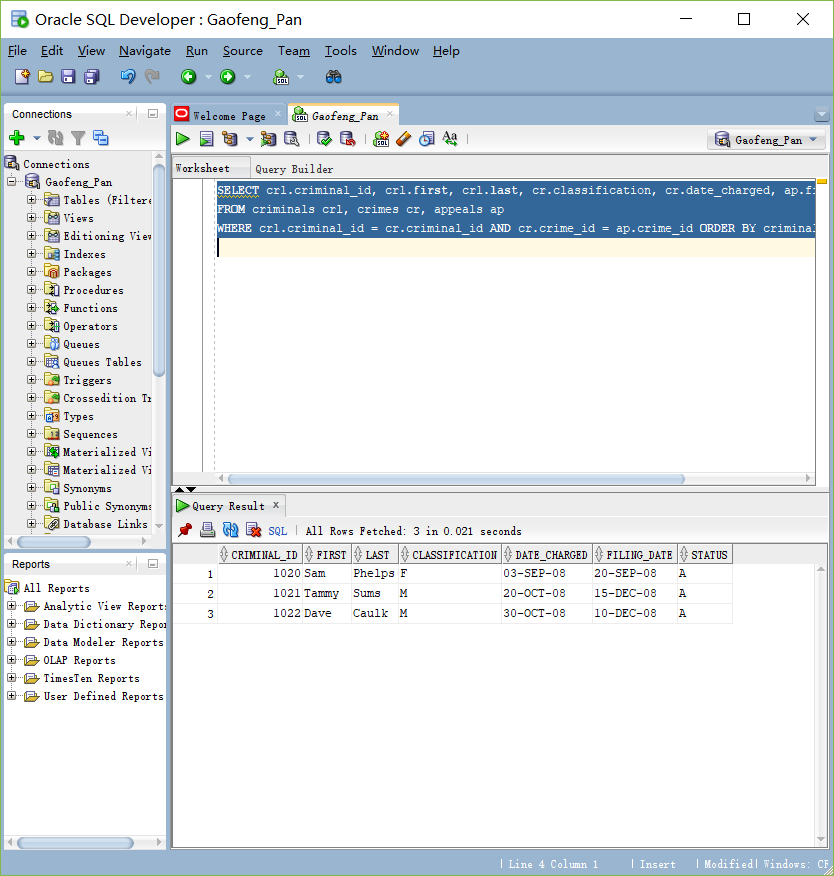
Traditional Method

SELECT crl.criminal\_id, crl.first, crl.last, cr.classification, cr.date\_charged, ap.filing\_date, ap.status

FROM criminals crl, crimes cr, appeals ap

WHERE crl.criminal\_id = cr.criminal\_id AND cr.crime\_id = ap.crime\_id

ORDER BY criminal\_id;



JOIN Method

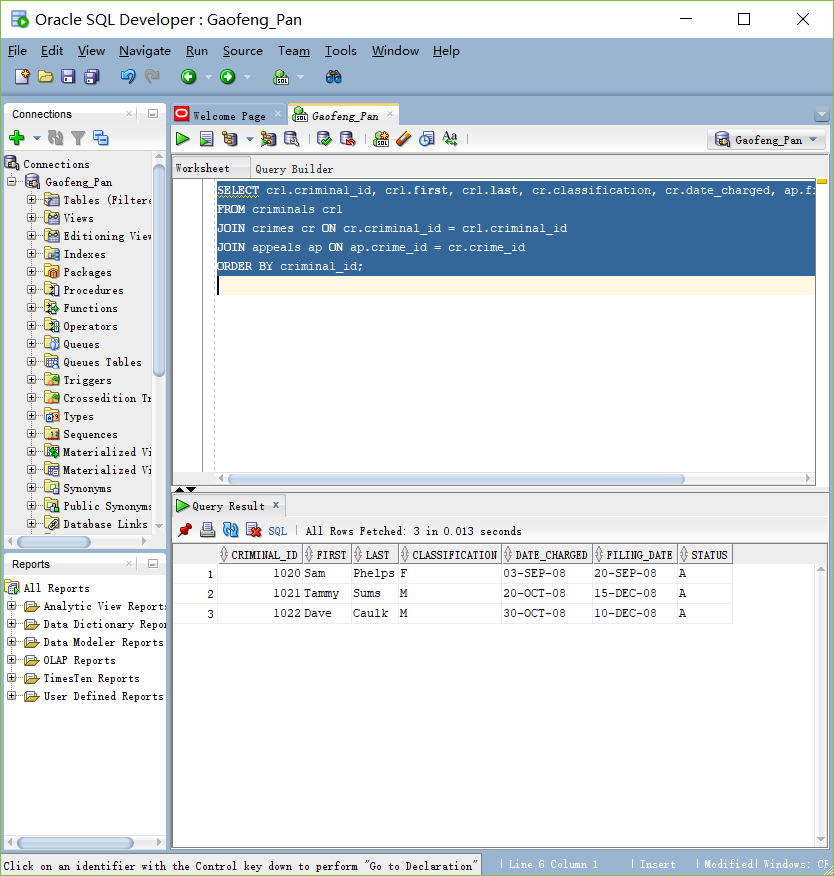
SELECT crl.criminal\_id, crl.first, crl.last, cr.classification, cr.date\_charged, ap.filing\_date, ap.status

FROM criminals crl

JOIN crimes cr ON cr.criminal\_id = crl.criminal\_id

JOIN appeals ap ON ap.crime\_id = cr.crime\_id

ORDER BY criminal\_id;



3. List all criminals along with crime information. The report needs to include the criminal ID,

name, crime classification, date charged, crime code, and fine amount. Include only crimes

classified as “Other.” Sort the list by criminal ID and date charged.

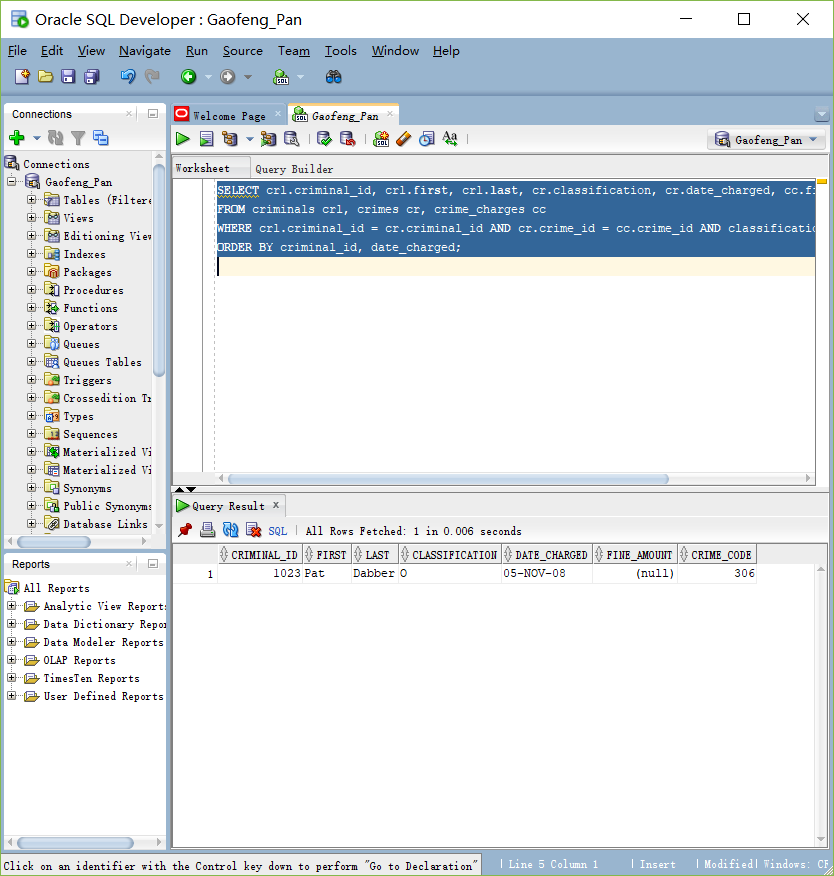
Traditional Method

SELECT crl.criminal\_id, crl.first, crl.last, cr.classification, cr.date\_charged, cc.fine\_amount, cc.crime\_code

FROM criminals crl, crimes cr, crime\_charges cc

WHERE crl.criminal\_id = cr.criminal\_id AND cr.crime\_id = cc.crime\_id AND classification = 'O'

ORDER BY criminal\_id, date\_charged;



JOIN METHOD

SELECT crl.criminal\_id, crl.first, crl.last, cr.classification, cr.date\_charged, cc.fine\_amount, cc.crime\_code

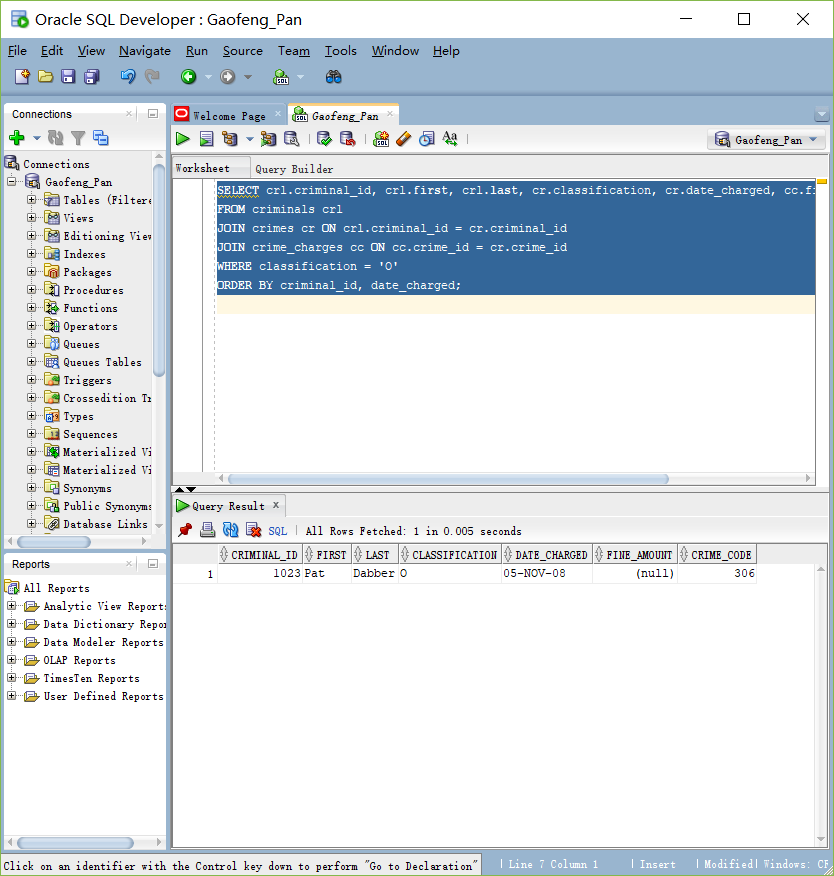
FROM criminals crl

JOIN crimes cr ON crl.criminal\_id = cr.criminal\_id

JOIN crime\_charges cc ON cc.crime\_id = cr.crime\_id

WHERE classification = 'O'

ORDER BY criminal\_id, date\_charged;



4. Create an alphabetical list of all criminals, including criminal ID, name, violent offender

status, parole status, and any known aliases.

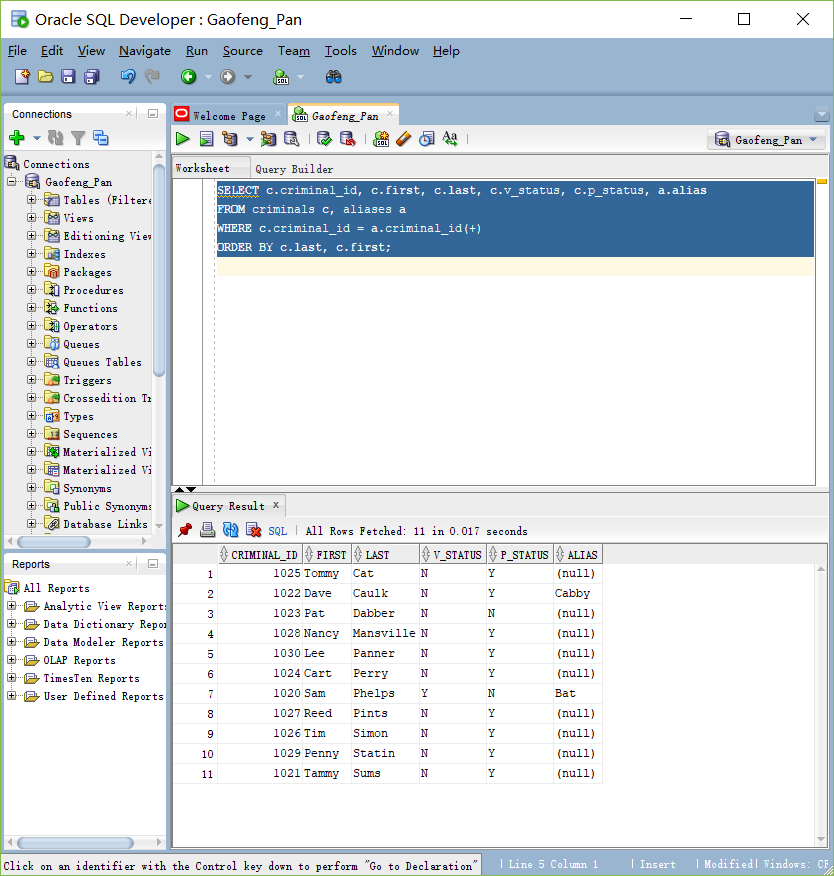
Traditional Method

SELECT c.criminal\_id, c.first, c.last, c.v\_status, c.p\_status, a.alias

FROM criminals c, aliases a

WHERE c.criminal\_id = a.criminal\_id(+)

ORDER BY c.last, c.first;



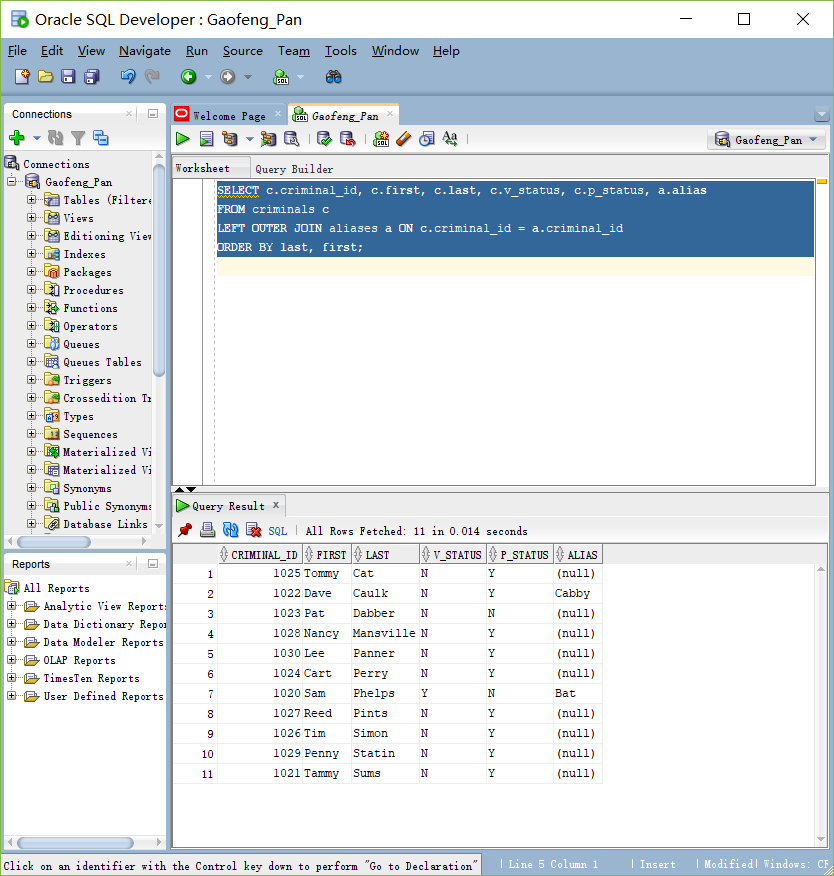
JOIN Method

SELECT c.criminal\_id, c.first, c.last, c.v\_status, c.p\_status, a.alias

FROM criminals c

LEFT OUTER JOIN aliases a ON c.criminal\_id = a.criminal\_id

ORDER BY last, first;



5. A table named Prob\_Contact contains the required frequency of contact with a probation

officer, based on the length of the probation period (the number of days assigned to probation).

Review the data in this table, which indicates ranges for the number of days and

applicable contact frequencies. Create a list containing the name of each criminal who has

been assigned a probation period, which is indicated by the sentence type. The list should

contain the criminal name, probation start date, probation end date, and required frequency

of contact. Sort the list by criminal name and probation start date.

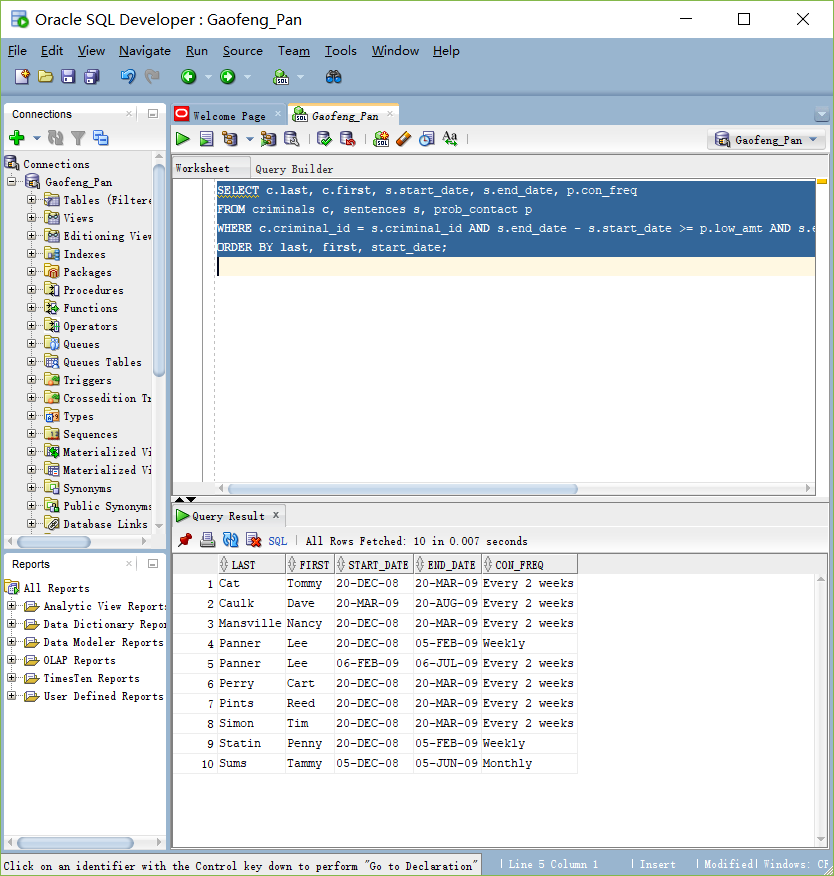
Traditional Method

SELECT c.last, c.first, s.start\_date, s.end\_date, p.con\_freq

FROM criminals c, sentences s, prob\_contact p

WHERE c.criminal\_id = s.criminal\_id AND s.end\_date - s.start\_date >= p.low\_amt AND s.end\_date - s.start\_date <= p.high\_amt AND s.type = 'P'

ORDER BY last, first, start\_date;



JOIN Method

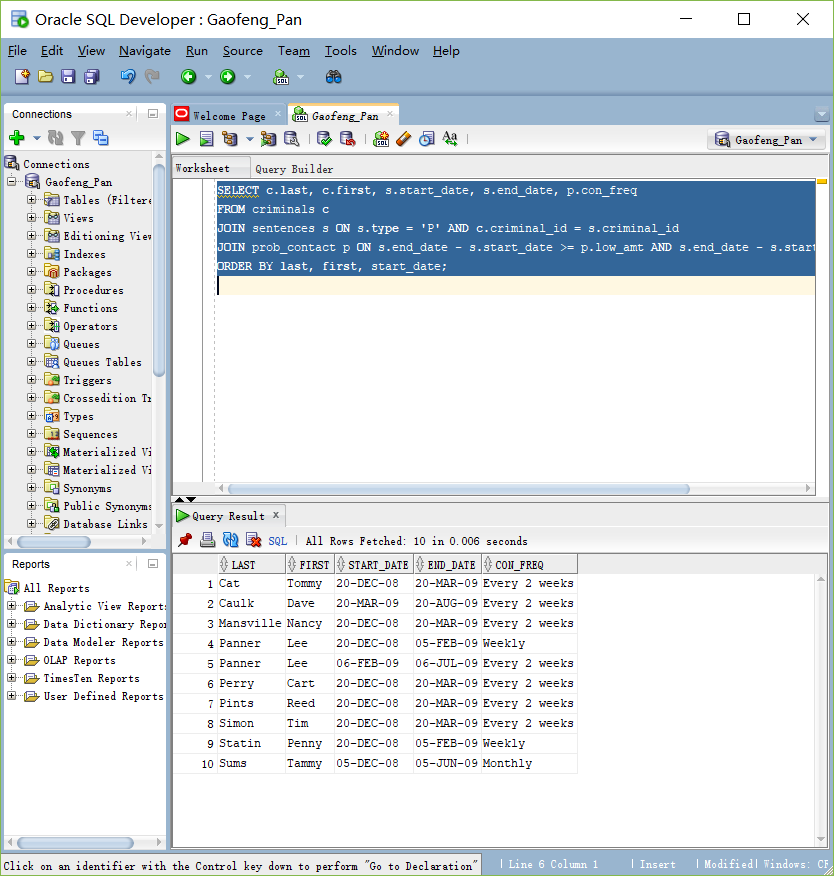
SELECT c.last, c.first, s.start\_date, s.end\_date, p.con\_freq

FROM criminals c

JOIN sentences s ON s.type = 'P' AND c.criminal\_id = s.criminal\_id

JOIN prob\_contact p ON s.end\_date - s.start\_date >= p.low\_amt AND s.end\_date - s.start\_date <= p.high\_amt

ORDER BY last, first, start\_date;



6. A column named Mgr\_ID has been added to the Prob\_Officers table and contains the ID

number of the probation supervisor for each officer. Produce a list showing each probation

officer’s name and his or her supervisor’s name. Sort the list alphabetically by probation

officer name.

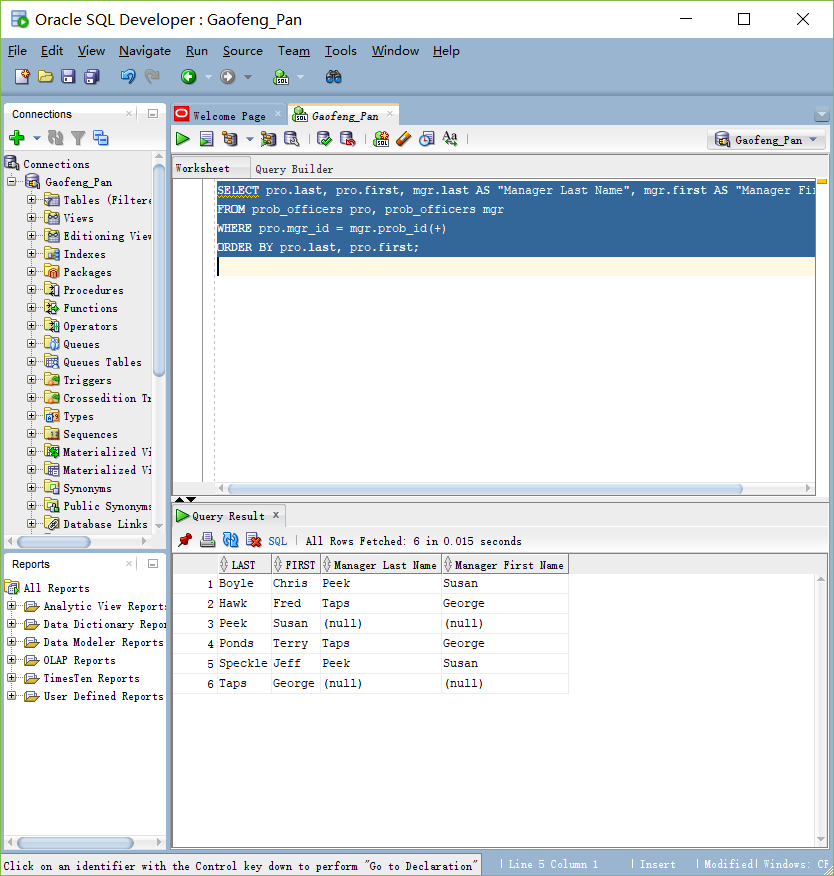
Tradtional Method

SELECT pro.last, pro.first, mgr.last AS "Manager Last Name", mgr.first AS "Manager First Name"

FROM prob\_officers pro, prob\_officers mgr

WHERE pro.mgr\_id = mgr.prob\_id(+)

ORDER BY pro.last, pro.first;



JOIN Method

SELECT pro.last, pro.first, mgr.last AS "Manager Last Name", mgr.first AS "Manager First Name"

FROM prob\_officers pro

LEFT OUTER JOIN prob\_officers mgr ON pro.mgr\_id = mgr.prob\_id

ORDER BY last, first;

