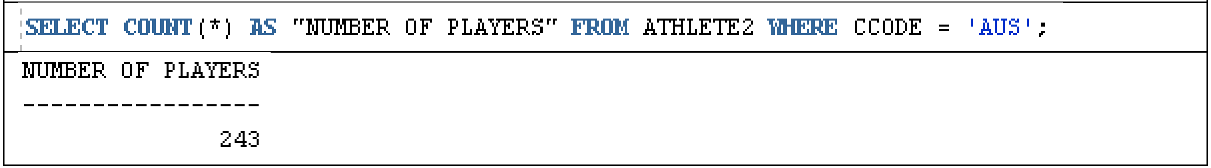
Harsha Joshi
45561623

Practical 1 Report

Task 1

Task 1.1

```
SELECT COUNT(*) AS "NUMBER OF PLAYERS" FROM ATHLETE2 WHERE CCODE = 'AUS';
```

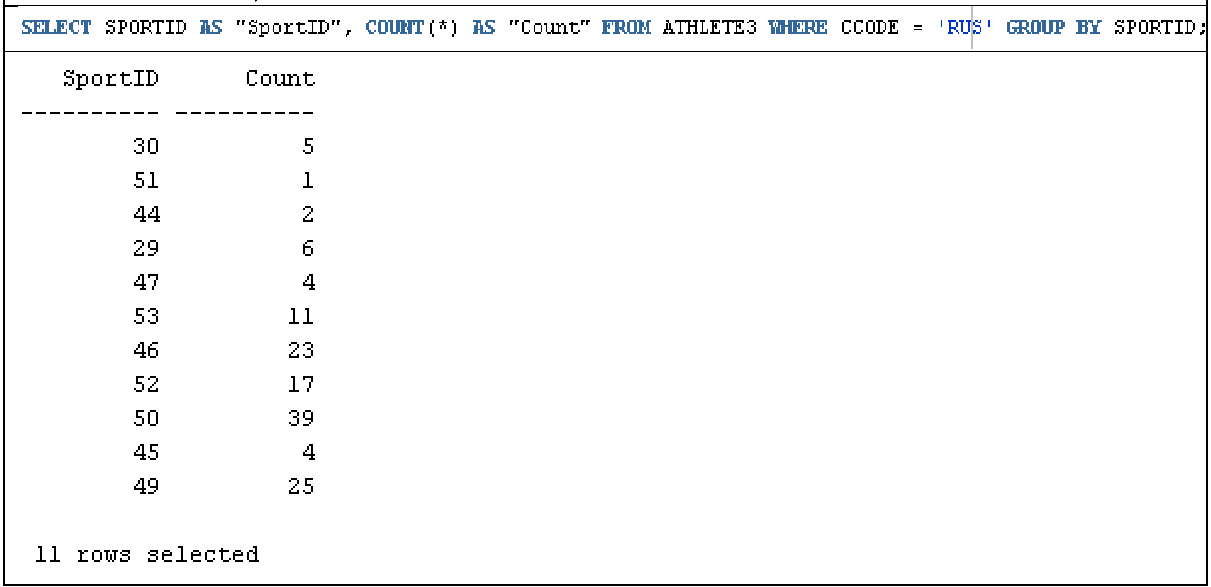


```
SELECT COUNT(*) AS "NUMBER OF PLAYERS" FROM ATHLETE2 WHERE CCODE = 'AUS';
```

```
NUMBER OF PLAYERS
-----------------
              243
```

Task 1.2

```
SELECT SPORTID AS "SportID", COUNT(*) AS "Count" FROM ATHLETE3 WHERE CCODE = 'RUS'
GROUP BY SPORTID;
```



```
SELECT SPORTID AS "SportID", COUNT(*) AS "Count" FROM ATHLETE3 WHERE CCODE = 'RUS' GROUP BY SPORTID;
```

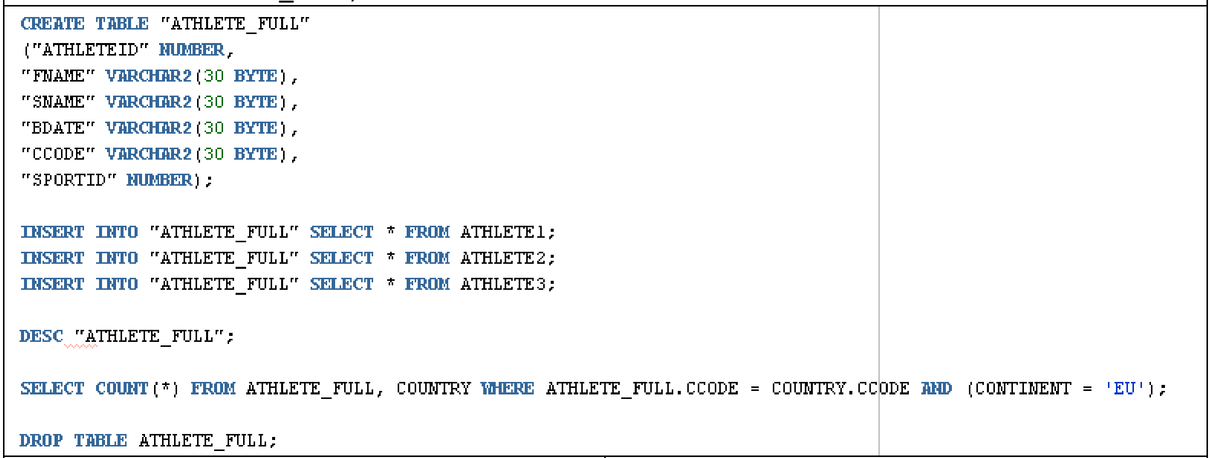
```
   SportID      Count
---------- ----------
        30          5
        51          1
        44          2
        29          6
        47          4
        53         11
        46         23
        52         17
        50         39
        45          4
        49         25

 11 rows selected
```

Task 1.3

```
CREATE TABLE "ATHLETE_FULL"
("ATHLETEID" NUMBER,
"FNAME" VARCHAR2(30 BYTE),
"SNAME" VARCHAR2(30 BYTE),
"BDATE" VARCHAR2(30 BYTE),
"CCODE" VARCHAR2(30 BYTE),
"SPORTID" NUMBER);

INSERT INTO "ATHLETE_FULL" SELECT * FROM ATHLETE1;
INSERT INTO "ATHLETE_FULL" SELECT * FROM ATHLETE2;
INSERT INTO "ATHLETE_FULL" SELECT * FROM ATHLETE3;

DESC "ATHLETE_FULL";
```

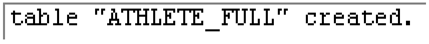
SELECT COUNT(*) FROM ATHLETE_FULL, COUNTRY WHERE ATHLETE_FULL.CCODE = COUNTRY.CCODE AND (CONTINENT = 'EU');

DROP TABLE ATHLETE_FULL;



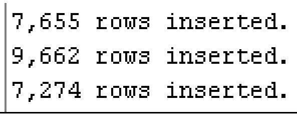
```
CREATE TABLE "ATHLETE_FULL"
("ATHLETEID" NUMBER,
"FNAME" VARCHAR2(30 BYTE),
"SNAME" VARCHAR2(30 BYTE),
"BDATE" VARCHAR2(30 BYTE),
"CCODE" VARCHAR2(30 BYTE),
"SPORTID" NUMBER);

INSERT INTO "ATHLETE_FULL" SELECT * FROM ATHLETE1;
INSERT INTO "ATHLETE_FULL" SELECT * FROM ATHLETE2;
INSERT INTO "ATHLETE_FULL" SELECT * FROM ATHLETE3;

DESC "ATHLETE_FULL";

SELECT COUNT(*) FROM ATHLETE_FULL, COUNTRY WHERE ATHLETE_FULL.CCODE = COUNTRY.CCODE AND (CONTINENT = 'EU');

DROP TABLE ATHLETE_FULL;
```

Query 1) Creating the ATHLETE_FULL table.



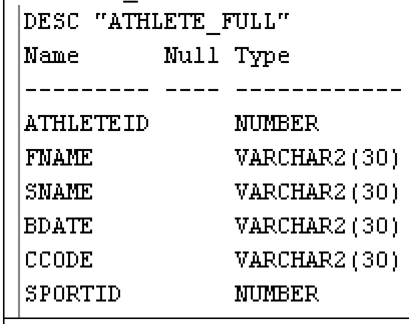
```
table "ATHLETE_FULL" created.
```

Query 2) Inserting the rows from ATHLETE1, ATHLETE2, ATHLETE3 respectively.



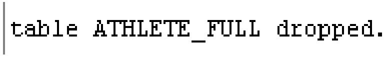
```
7,655 rows inserted.
9,662 rows inserted.
7,274 rows inserted.
```

Query 3) Displaying the attributes of ATHLETE_FULL table.



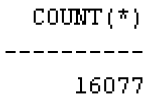
```
DESC "ATHLETE_FULL"
Name      Null Type
--------- ---- ------------
ATHLETEID      NUMBER
FNAME          VARCHAR2(30)
SNAME          VARCHAR2(30)
BDATE          VARCHAR2(30)
CCODE          VARCHAR2(30)
SPORTID        NUMBER
```

Query 5) Dropping the ATHLETE_FULL table to avoid duplication in the count.



```
table ATHLETE_FULL dropped.
```

Query 4) Demonstrating the count of the atheletes from Europe.

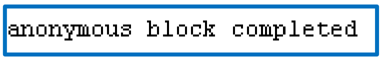


```
  COUNT(*)
----------
     16077
```

NOTE: The ATHLETE_FULL table is dropped at the end of the query due to the count being duplicated multiple times when the script is run. This is because every time the script is re-run the same tables (ATHLETE1, ATHLETE2, ATHLETE3) will be re-inserted into ATHLETE_FULL, thereby duplicating the count of athletes whose continent is Europe. It is important to drop ATHLETE_FULL so to ensure the accurate count.

Task 2

For all the following queries, BEGIN and END commands needed to be removed and only the nested query (Line 2-4 in all users below) in the user script called s4556162 was run to demonstrate the output screenshotted. Otherwise, the following appears and does not specify whether the ATHLETEID = 305 row, has been updated:

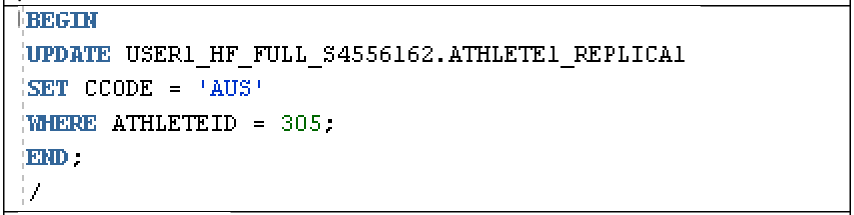


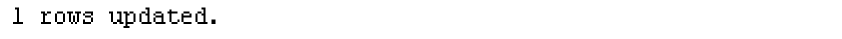
```
anonymous block completed
```

This is because the nested/executable part of the query is written within a code block. When the message "anonymous block completed" shows up, it means that there are no errors within the nested query (between the BEGIN and END commands) and the code block has successfully executed. To confirm that the row where ATHLETEID = 305 has actually been updated, we must remove the BEGIN and END commands (Line 1, 5, 6 in all users below). The BEGIN and END commands were left in the code and screenshots below because this is the standard SQL practice to update a database table (Oracle, 2022).

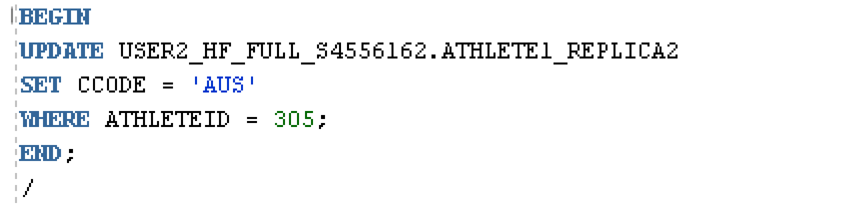
Job 1 – Full Replication	

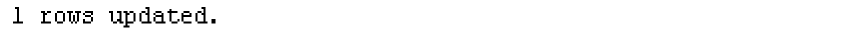
USER1_HF_FULL_S4556162	BEGIN
	UPDATE USER1_HF_FULL_S4556162.ATHLETE1_REPLICA1
	SET CCODE = 'AUS'
	WHERE ATHLETEID = 305;
	END;
	/

	
	BEGIN
	UPDATE USER1_HF_FULL_S4556162.ATHLETE1_REPLICA1
	SET CCODE = 'AUS'
	WHERE ATHLETEID = 305;
	END;
	/

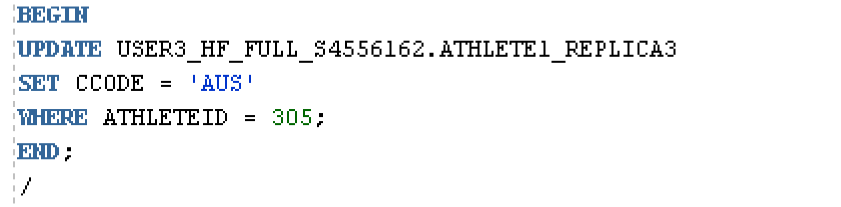
	 1 rows updated.

USER2_HF_FULL_S4556162	BEGIN
	UPDATE USER2_HF_FULL_S4556162.ATHLETE1_REPLICA2
	SET CCODE = 'AUS'
	WHERE ATHLETEID = 305;
	END;
	/

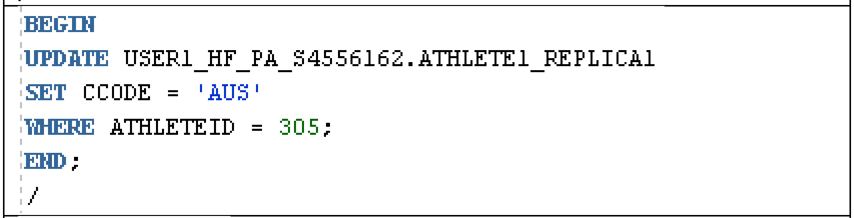
	
	BEGIN
	UPDATE USER2_HF_FULL_S4556162.ATHLETE1_REPLICA2
	SET CCODE = 'AUS'
	WHERE ATHLETEID = 305;
	END;
	/

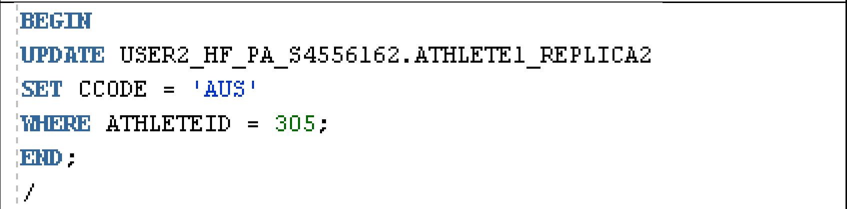
	 1 rows updated.

USER3_HF_FULL_S4556162	BEGIN
	UPDATE USER3_HF_FULL_S4556162.ATHLETE1_REPLICA3
	SET CCODE = 'AUS'
	WHERE ATHLETEID = 305;
	END;
	/

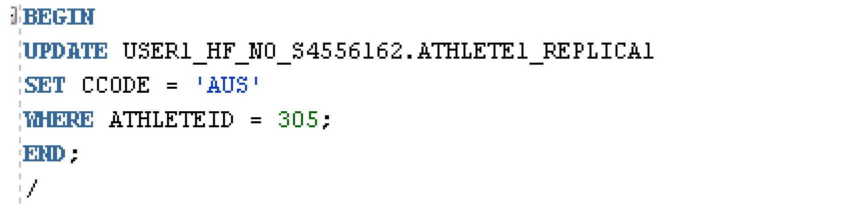
	
	BEGIN
	UPDATE USER3_HF_FULL_S4556162.ATHLETE1_REPLICA3
	SET CCODE = 'AUS'
	WHERE ATHLETEID = 305;
	END;
	/
	
	1 rows updated.

Job 2 – Partial Replication	

USER1_HF_PA_S4556162	BEGIN
	UPDATE USER1_HF_PA_S4556162.ATHLETE1_REPLICA1
	SET CCODE = 'AUS'
	WHERE ATHLETEID = 305;
	END;
	/
	
	BEGIN
	UPDATE USER1_HF_PA_S4556162.ATHLETE1_REPLICA1
	SET CCODE = 'AUS'
	WHERE ATHLETEID = 305;
	END;
	/
	
	1 rows updated.

USER2_HF_ PA _S4556162	BEGIN
	UPDATE USER2_HF_PA_S4556162.ATHLETE1_REPLICA2
	SET CCODE = 'AUS'
	WHERE ATHLETEID = 305;
	END;
	/
	
	BEGIN
	UPDATE USER2_HF_PA_S4556162.ATHLETE1_REPLICA2
	SET CCODE = 'AUS'
	WHERE ATHLETEID = 305;
	END;
	/
	
	1 rows updated.

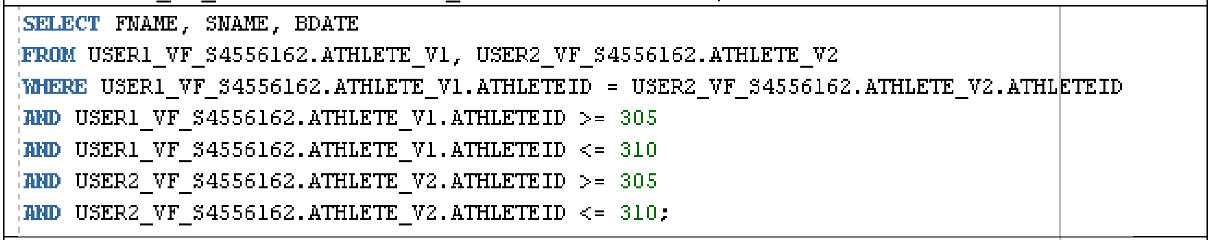
USER3_HF_ PA _S4556162	Pass – does not need to be updated as this database user only contains the following tables: • ATHLETE2_REPLICA2 • ATHLETE3_REPLICA3
Job 3 – No Replication	

USER1_HF_NO_S4556162	BEGIN
	UPDATE USER1_HF_NO_S4556162.ATHLETE1_REPLICA1
	SET CCODE = 'AUS'
	WHERE ATHLETEID = 305;
	END;
	/
	
	BEGIN
	UPDATE USER1_HF_NO_S4556162.ATHLETE1_REPLICA1
	SET CCODE = 'AUS'
	WHERE ATHLETEID = 305;
	END;
	/

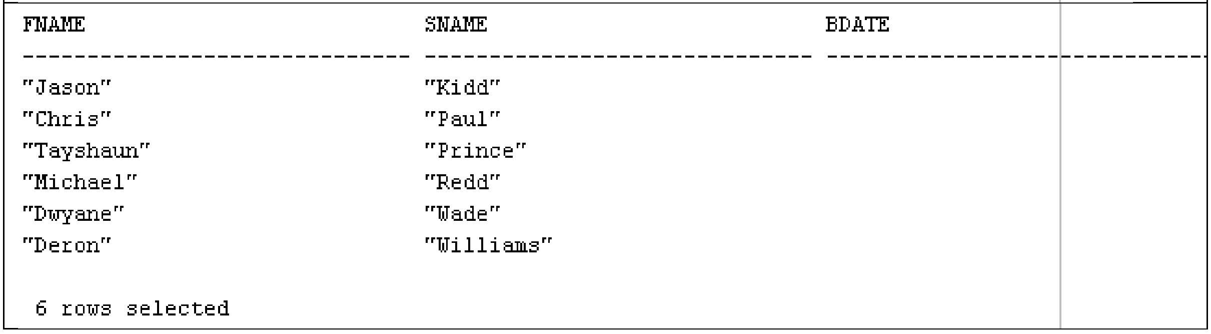
	`1 rows updated.`
USER2_HF_ NO _S4556162	Pass – does not need to be updated as this database user only contains the following tables: • ATHLETE2_REPLICA2
USER3_HF_ NO _S4556162	Pass – does not need to be updated as this database user only contains the following tables: • ATHLETE3_REPLICA3

Task 3

```
SELECT FNAME, SNAME, BDATE
FROM USER1_VF_S4556162.ATHLETE_V1, USER2_VF_S4556162.ATHLETE_V2
WHERE
USER1_VF_S4556162.ATHLETE_V1.ATHLETEID = USER2_VF_S4556162.ATHLETE_V2.ATHLETEID
AND USER1_VF_S4556162.ATHLETE_V1.ATHLETEID >= 305
AND USER1_VF_S4556162.ATHLETE_V1.ATHLETEID <= 310
AND USER2_VF_S4556162.ATHLETE_V2.ATHLETEID >= 305
AND USER2_VF_S4556162.ATHLETE_V2.ATHLETEID <= 310;
```



```
SELECT FNAME, SNAME, BDATE
FROM USER1_VF_S4556162.ATHLETE_V1, USER2_VF_S4556162.ATHLETE_V2
WHERE USER1_VF_S4556162.ATHLETE_V1.ATHLETEID = USER2_VF_S4556162.ATHLETE_V2.ATHLETEID
AND USER1_VF_S4556162.ATHLETE_V1.ATHLETEID >= 305
AND USER1_VF_S4556162.ATHLETE_V1.ATHLETEID <= 310
AND USER2_VF_S4556162.ATHLETE_V2.ATHLETEID >= 305
AND USER2_VF_S4556162.ATHLETE_V2.ATHLETEID <= 310;
```



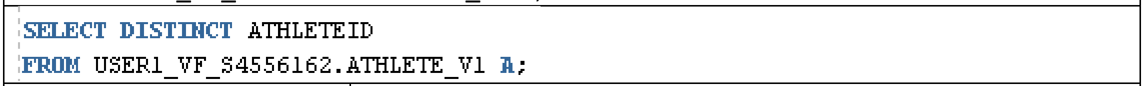
```
FNAME                          SNAME                          BDATE
------------------------------ ------------------------------ ------------------------------
"Jason"                        "Kidd"
"Chris"                        "Paul"
"Tayshaun"                     "Prince"
"Michael"                      "Redd"
"Dwyane"                       "Wade"
"Deron"                        "Williams"

 6 rows selected
```

Task 4

Semi Joins

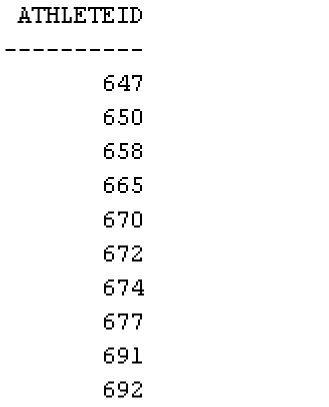
STEP 1

SELECT DISTINCT ATHLETEID
FROM USER1_VF_S4556162.ATHLETE_V1 A;



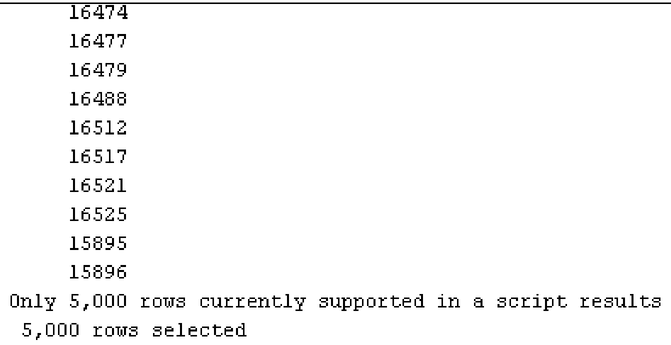
```
SELECT DISTINCT ATHLETEID
FROM USER1_VF_S4556162.ATHLETE_V1 A;
```

First 10 Outputs



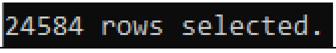
```
 ATHLETEID
----------
       647
       650
       658
       665
       670
       672
       674
       677
       691
       692
```

Final 10 Outputs



```
     16474
     16477
     16479
     16488
     16512
     16517
     16521
     16525
     15895
     15896
Only 5,000 rows currently supported in a script results
 5,000 rows selected
```

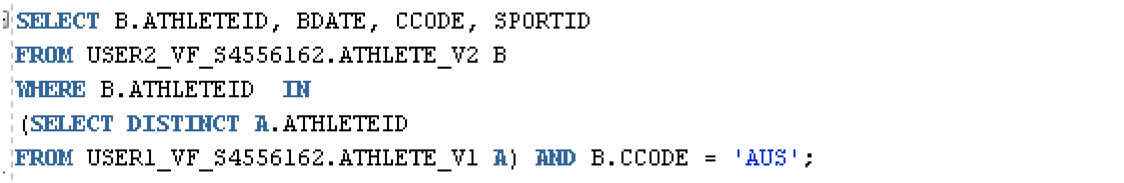
NOTE: 5000 rows supported and displayed (in SQL Developer), actual output was 248584 rows (according to SQL Plus, specified in calculations below)



```
24584 rows selected.
```

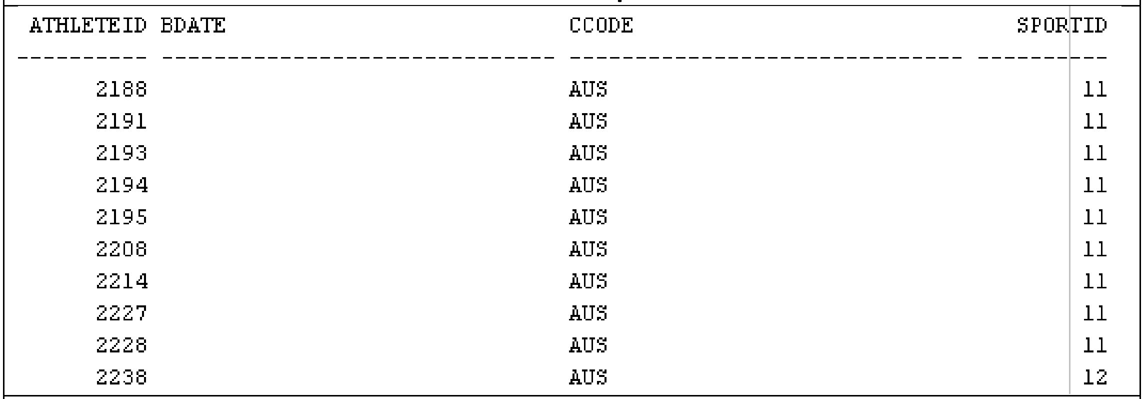
STEP 2

SELECT B.ATHLETEID, BDATE, CCODE, SPORTID
FROM USER2_VF_S4556162.ATHLETE_V2 B
WHERE B.ATHLETEID IN
(SELECT DISTINCT A.ATHLETEID
FROM USER1_VF_S4556162.ATHLETE_V1 A) AND B.CCODE = 'AUS';



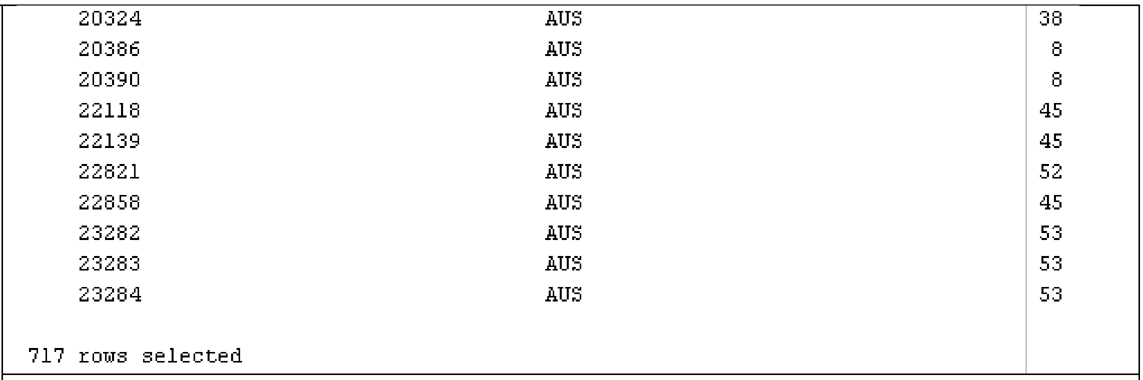
```
SELECT B.ATHLETEID, BDATE, CCODE, SPORTID
FROM USER2_VF_S4556162.ATHLETE_V2 B
WHERE B.ATHLETEID  IN
(SELECT DISTINCT A.ATHLETEID
FROM USER1_VF_S4556162.ATHLETE_V1 A) AND B.CCODE = 'AUS';
```

First 10 Outputs



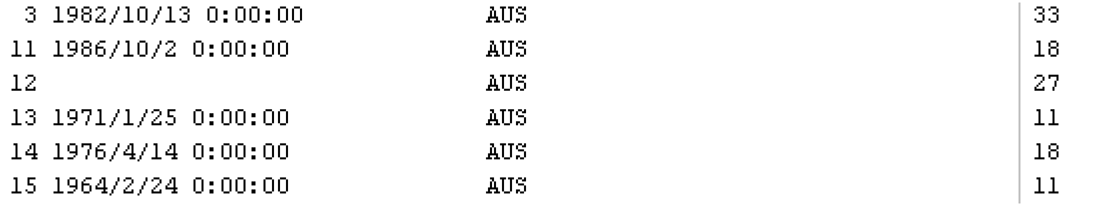
```
 ATHLETEID BDATE                          CCODE                             SPORTID
---------- ------------------------------ ------------------------------ ----------
      2188                                AUS                                    11
      2191                                AUS                                    11
      2193                                AUS                                    11
      2194                                AUS                                    11
      2195                                AUS                                    11
      2208                                AUS                                    11
      2214                                AUS                                    11
      2227                                AUS                                    11
      2228                                AUS                                    11
      2238                                AUS                                    12
```

Final 10 Outputs



```
    20324                          AUS                    38
    20386                          AUS                     8
    20390                          AUS                     8
    22118                          AUS                    45
    22139                          AUS                    45
    22821                          AUS                    52
    22858                          AUS                    45
    23282                          AUS                    53
    23283                          AUS                    53
    23284                          AUS                    53

717 rows selected
```

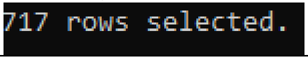
Please Note: BDATE was visible for some of the other entries (besides the first/final 10 outputs) in which the field was specified. The below screenshots demonstrate some examples:



```
 3 1982/10/13 0:00:00          AUS                    33
11 1986/10/2 0:00:00           AUS                    18
12                             AUS                    27
13 1971/1/25 0:00:00           AUS                    11
14 1976/4/14 0:00:00           AUS                    18
15 1964/2/24 0:00:00           AUS                    11
```

These are consistent with the rest of the database's raw data provided.

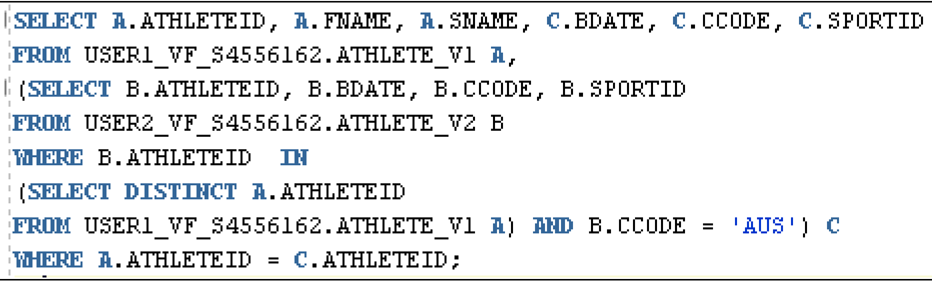
NOTE: 717 rows displayed, actual output was 717 rows (according to both SQL Plus & SQL Developer, specified in calculations below)



```
717 rows selected.
```

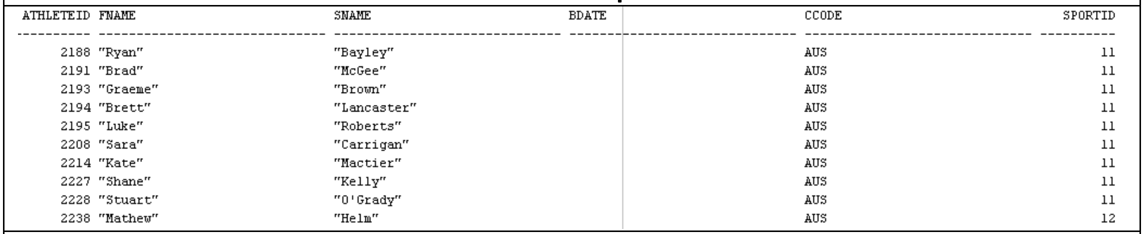
STEP 3

SELECT A.ATHLETEID, A.FNAME, A.SNAME, C.BDATE, C.CCODE, C.SPORTID
FROM USER1_VF_S4556162.ATHLETE_V1 A,
(SELECT B.ATHLETEID, B.BDATE, B.CCODE, B.SPORTID
FROM USER2_VF_S4556162.ATHLETE_V2 B
WHERE B.ATHLETEID IN
(SELECT DISTINCT A.ATHLETEID
FROM USER1_VF_S4556162.ATHLETE_V1 A) AND B.CCODE = 'AUS') C
WHERE A.ATHLETEID = C.ATHLETEID;



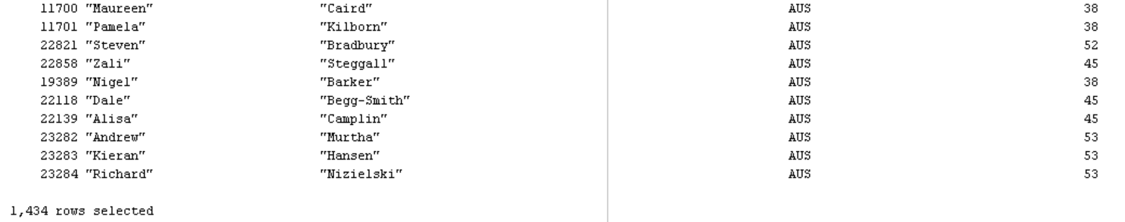
```
SELECT A.ATHLETEID, A.FNAME, A.SNAME, C.BDATE, C.CCODE, C.SPORTID
FROM USER1_VF_S4556162.ATHLETE_V1 A,
 (SELECT B.ATHLETEID, B.BDATE, B.CCODE, B.SPORTID
FROM USER2_VF_S4556162.ATHLETE_V2 B
WHERE B.ATHLETEID  IN
 (SELECT DISTINCT A.ATHLETEID
FROM USER1_VF_S4556162.ATHLETE_V1 A) AND B.CCODE = 'AUS') C
WHERE A.ATHLETEID = C.ATHLETEID;
```

First 10 Outputs



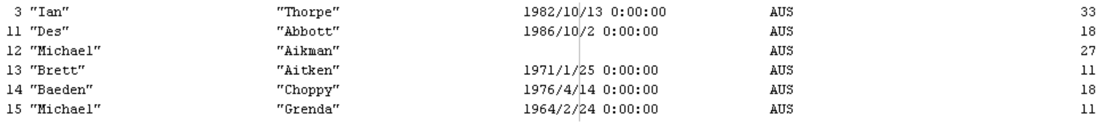
```
ATHLETEID FNAME                          SNAME                          BDATE                               CCODE                          SPORTID
---------- ------------------------------ ------------------------------ ----------------------------------- ------------------------------ ----------
      2188 "Ryan"                         "Bayley"                                                           AUS                                    11
      2191 "Brad"                         "McGee"                                                            AUS                                    11
      2193 "Graeme"                       "Brown"                                                            AUS                                    11
      2194 "Brett"                        "Lancaster"                                                        AUS                                    11
      2195 "Luke"                         "Roberts"                                                          AUS                                    11
      2208 "Sara"                         "Carrigan"                                                         AUS                                    11
      2214 "Kate"                         "Mactier"                                                          AUS                                    11
      2227 "Shane"                        "Kelly"                                                            AUS                                    11
      2228 "Stuart"                       "O'Grady"                                                          AUS                                    11
      2238 "Mathew"                       "Helm"                                                             AUS                                    12
```

Final 10 Outputs



```
     11700 "Maureen"                      "Caird"                                                      AUS                        38
     11701 "Pamela"                       "Kilborn"                                                    AUS                        38
     22821 "Steven"                       "Bradbury"                                                   AUS                        52
     22858 "Zali"                         "Steggall"                                                   AUS                        45
     19389 "Nigel"                        "Barker"                                                     AUS                        38
     22118 "Dale"                         "Begg-Smith"                                                 AUS                        45
     22139 "Alisa"                        "Camplin"                                                    AUS                        45
     23282 "Andrew"                       "Murtha"                                                     AUS                        53
     23283 "Kieran"                       "Hansen"                                                     AUS                        53
     23284 "Richard"                      "Nizielski"                                                  AUS                        53

1,434 rows selected
```

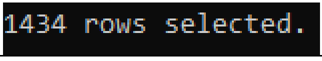
Please Note: BDATE was visible for some of the other entries (besides the first/final 10 outputs) in which the field was specified. The below screenshots demonstrate some examples:



```
 3 "Ian"                  "Thorpe"              1982/10/13 0:00:00        AUS                        33
11 "Des"                  "Abbott"              1986/10/2 0:00:00         AUS                        18
12 "Michael"              "Aikman"                                        AUS                        27
13 "Brett"                "Aitken"              1971/1/25 0:00:00         AUS                        11
14 "Baeden"               "Choppy"              1976/4/14 0:00:00         AUS                        18
15 "Michael"              "Grenda"              1964/2/24 0:00:00         AUS                        11
```

These are consistent with the rest of the database's raw data provided.

NOTE: 1434 rows displayed, actual output was 1434 rows (according to both SQL Plus & SQL Developer, specified in calculations below)

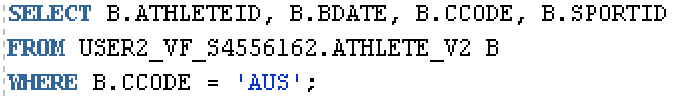


```
1434 rows selected.
```

Inner Joins

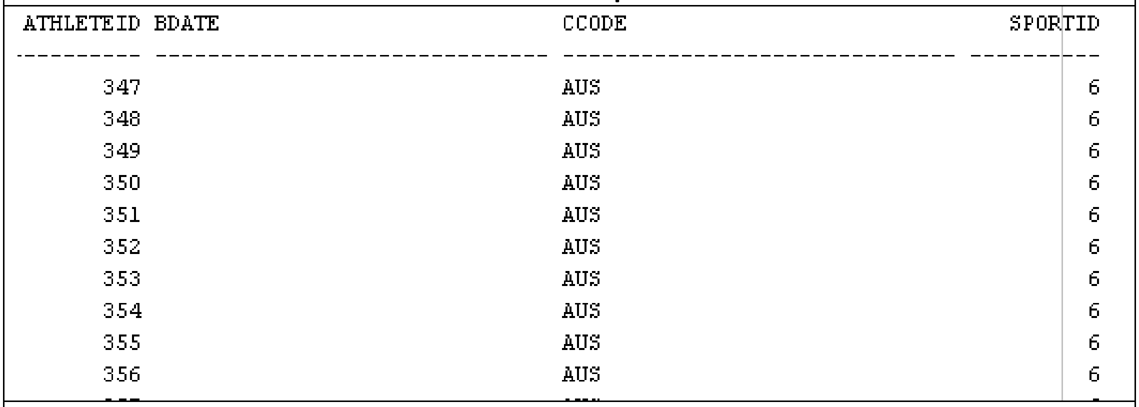
STEP 1

SELECT B.ATHLETEID, B.BDATE, B.CCODE, B.SPORTID
FROM USER2_VF_S4556162.ATHLETE_V2 B
WHERE B.CCODE = 'AUS';



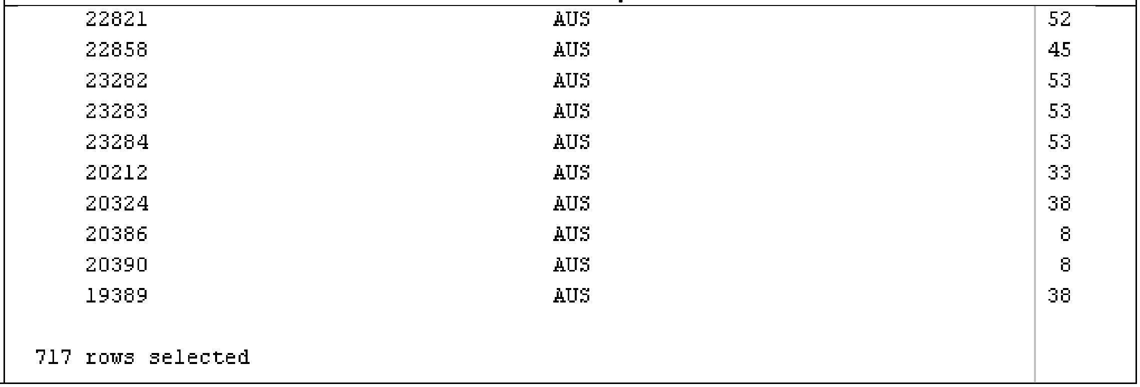
```
SELECT B.ATHLETEID, B.BDATE, B.CCODE, B.SPORTID
FROM USER2_VF_S4556162.ATHLETE_V2 B
WHERE B.CCODE = 'AUS';
```

First 10 Outputs



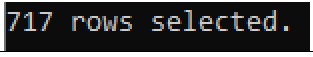
```
ATHLETEID BDATE                          CCODE                             SPORTID
---------- ------------------------------ ------------------------------ ----------
       347                                AUS                                     6
       348                                AUS                                     6
       349                                AUS                                     6
       350                                AUS                                     6
       351                                AUS                                     6
       352                                AUS                                     6
       353                                AUS                                     6
       354                                AUS                                     6
       355                                AUS                                     6
       356                                AUS                                     6
```

Final 10 Outputs



```
     22821                               AUS                              52
     22858                               AUS                              45
     23282                               AUS                              53
     23283                               AUS                              53
     23284                               AUS                              53
     20212                               AUS                              33
     20324                               AUS                              38
     20386                               AUS                               8
     20390                               AUS                               8
     19389                               AUS                              38

 717 rows selected
```

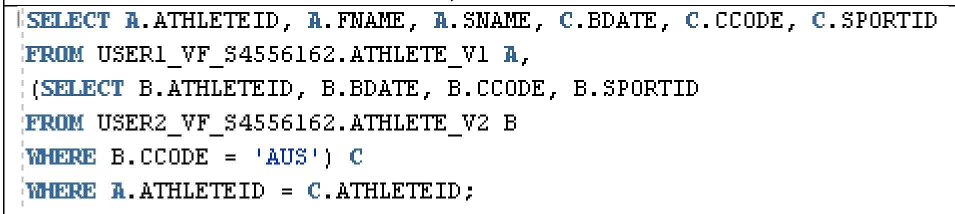
NOTE: 717 rows displayed, actual output was 717 rows (according to both SQL Plus & SQL Developer, specified in calculations below)



```
717 rows selected.
```

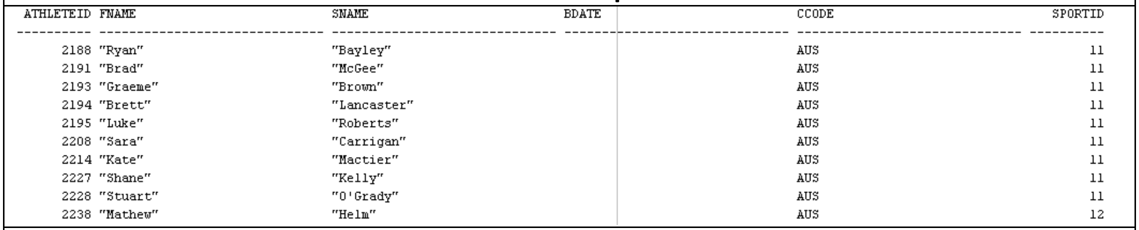
STEP 2

SELECT A.ATHLETEID, A.FNAME, A.SNAME, C.BDATE, C.CCODE, C.SPORTID
FROM USER1_VF_S4556162.ATHLETE_V1 A,
(SELECT B.ATHLETEID, B.BDATE, B.CCODE, B.SPORTID
FROM USER2_VF_S4556162.ATHLETE_V2 B
WHERE B.CCODE = 'AUS') C
WHERE A.ATHLETEID = C.ATHLETEID;



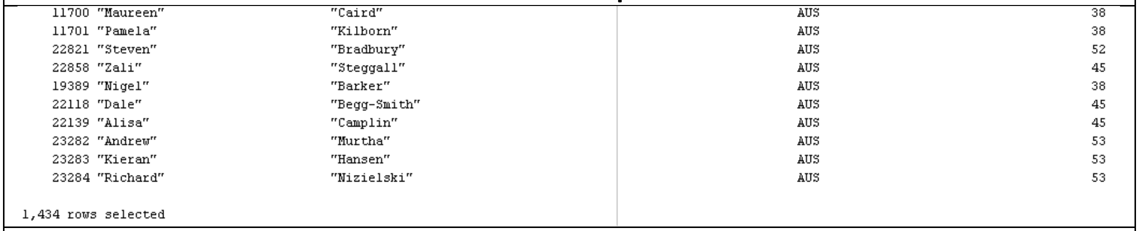
```
SELECT A.ATHLETEID, A.FNAME, A.SNAME, C.BDATE, C.CCODE, C.SPORTID
FROM USER1_VF_S4556162.ATHLETE_V1 A,
 (SELECT B.ATHLETEID, B.BDATE, B.CCODE, B.SPORTID
FROM USER2_VF_S4556162.ATHLETE_V2 B
WHERE B.CCODE = 'AUS') C
WHERE A.ATHLETEID = C.ATHLETEID;
```

First 10 Outputs



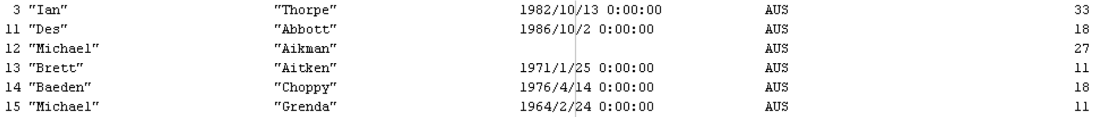
```
ATHLETEID FNAME                          SNAME                          BDATE                          CCODE                          SPORTID
---------- ------------------------------ ------------------------------ ------------------------------ ------------------------------ ----------
      2188 "Ryan"                         "Bayley"                                                      AUS                                    11
      2191 "Brad"                         "McGee"                                                       AUS                                    11
      2193 "Graeme"                       "Brown"                                                       AUS                                    11
      2194 "Brett"                        "Lancaster"                                                   AUS                                    11
      2195 "Luke"                         "Roberts"                                                     AUS                                    11
      2208 "Sara"                         "Carrigan"                                                    AUS                                    11
      2214 "Kate"                         "Mactier"                                                     AUS                                    11
      2227 "Shane"                        "Kelly"                                                       AUS                                    11
      2228 "Stuart"                       "O'Grady"                                                     AUS                                    11
      2238 "Mathew"                       "Helm"                                                        AUS                                    12
```

Final 10 Outputs



```
     11700 "Maureen"                      "Caird"                                                       AUS                                    38
     11701 "Pamela"                       "Kilborn"                                                     AUS                                    38
     22821 "Steven"                       "Bradbury"                                                    AUS                                    52
     22858 "Zali"                         "Steggall"                                                    AUS                                    45
     19389 "Nigel"                        "Barker"                                                      AUS                                    38
     22118 "Dale"                         "Begg-Smith"                                                  AUS                                    45
     22139 "Alisa"                        "Camplin"                                                     AUS                                    45
     23282 "Andrew"                       "Murtha"                                                      AUS                                    53
     23283 "Kieran"                       "Hansen"                                                      AUS                                    53
     23284 "Richard"                      "Nizielski"                                                   AUS                                    53

1,434 rows selected
```

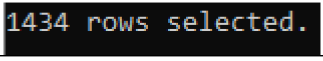
Please Note: BDATE was visible for some of the other entries (besides the first/final 10 outputs) in which the field was specified. The below screenshots demonstrate some examples:



```
 3 "Ian"                  "Thorpe"                 1982/10/13 0:00:00       AUS                      33
11 "Des"                  "Abbott"                 1986/10/2 0:00:00        AUS                      18
12 "Michael"              "Aikman"                                          AUS                      27
13 "Brett"                "Aitken"                 1971/1/25 0:00:00        AUS                      11
14 "Baeden"               "Choppy"                 1976/4/14 0:00:00        AUS                      18
15 "Michael"              "Grenda"                 1964/2/24 0:00:00        AUS                      11
```

These are consistent with the rest of the database's raw data provided.

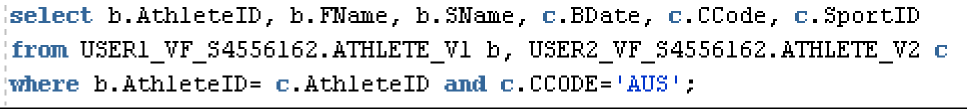
NOTE: 1434 rows displayed, actual output was 1434 rows (according to both SQL Plus & SQL Developer, specified in calculations below)



```
1434 rows selected.
```

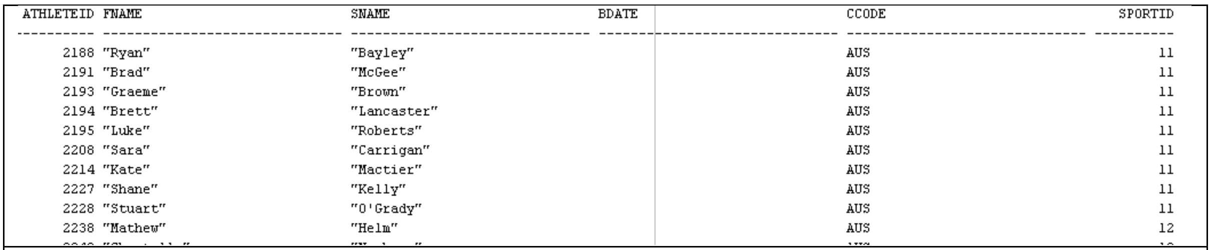
Global Join Query – Given in Practical 1 Task Sheet

select b.AthleteID, b.FName, b.SName, c.BDate, c.CCode, c.SportID
from USER1_VF_S4556162.ATHLETE_V1 b, USER2_VF_S4556162.ATHLETE_V2 c
where b.AthleteID= c.AthleteID and c.CCODE='AUS';



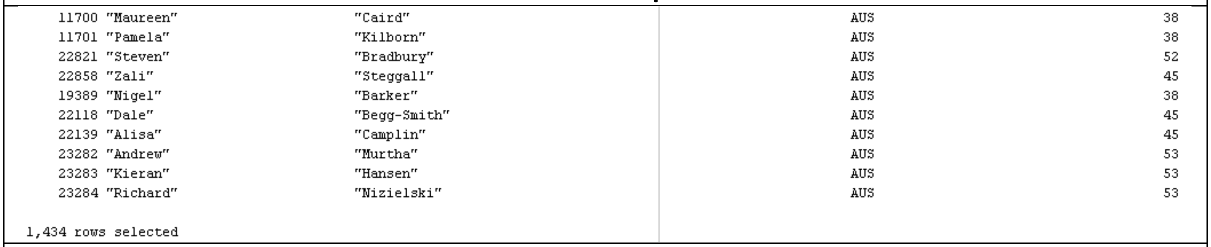
```
select b.AthleteID, b.FName, b.SName, c.BDate, c.CCode, c.SportID
from USER1_VF_S4556162.ATHLETE_V1 b, USER2_VF_S4556162.ATHLETE_V2 c
where b.AthleteID= c.AthleteID and c.CCODE='AUS';
```

First 10 Outputs



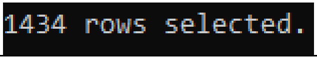
```
ATHLETEID FNAME                          SNAME                          BDATE                          CCODE                          SPORTID
---------- ------------------------------ ------------------------------ ------------------------------ ------------------------------ ----------
      2188 "Ryan"                         "Bayley"                                                     AUS                                    11
      2191 "Brad"                         "McGee"                                                      AUS                                    11
      2193 "Graeme"                       "Brown"                                                      AUS                                    11
      2194 "Brett"                        "Lancaster"                                                  AUS                                    11
      2195 "Luke"                         "Roberts"                                                    AUS                                    11
      2208 "Sara"                         "Carrigan"                                                   AUS                                    11
      2214 "Kate"                         "Mactier"                                                    AUS                                    11
      2227 "Shane"                        "Kelly"                                                      AUS                                    11
      2228 "Stuart"                       "O'Grady"                                                    AUS                                    11
      2238 "Mathew"                       "Helm"                                                       AUS                                    12
```

Final 10 Outputs



```
     11700 "Maureen"                      "Caird"                                                      AUS                                    38
     11701 "Pamela"                       "Kilborn"                                                    AUS                                    38
     22821 "Steven"                       "Bradbury"                                                   AUS                                    52
     22858 "Zali"                         "Steggall"                                                   AUS                                    45
     19389 "Nigel"                        "Barker"                                                     AUS                                    38
     22118 "Dale"                         "Begg-Smith"                                                 AUS                                    45
     22139 "Alisa"                        "Camplin"                                                    AUS                                    45
     23282 "Andrew"                       "Murtha"                                                     AUS                                    53
     23283 "Kieran"                       "Hansen"                                                     AUS                                    53
     23284 "Richard"                      "Nizielski"                                                  AUS                                    53

1,434 rows selected
```

NOTE: 1434 rows displayed, actual output was 1434 rows (according to both SQL Plus & SQL Developer, specified in calculations below)



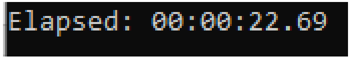
```
1434 rows selected.
```

Calculations

Semi Joins

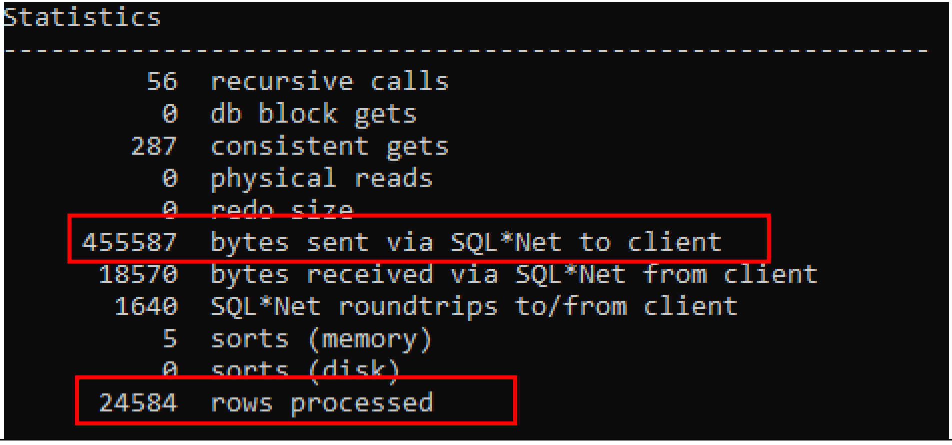
STEP 1

Time Elapsed:



```
Elapsed: 00:00:22.69
```

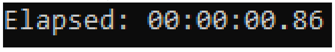
Statistical Output:



```
Statistics
----------------------------------------------------------
         56  recursive calls
          0  db block gets
        287  consistent gets
          0  physical reads
          0  redo size
     455587  bytes sent via SQL*Net to client
      18570  bytes received via SQL*Net from client
       1640  SQL*Net roundtrips to/from client
          5  sorts (memory)
          0  sorts (disk)
      24584  rows processed
```

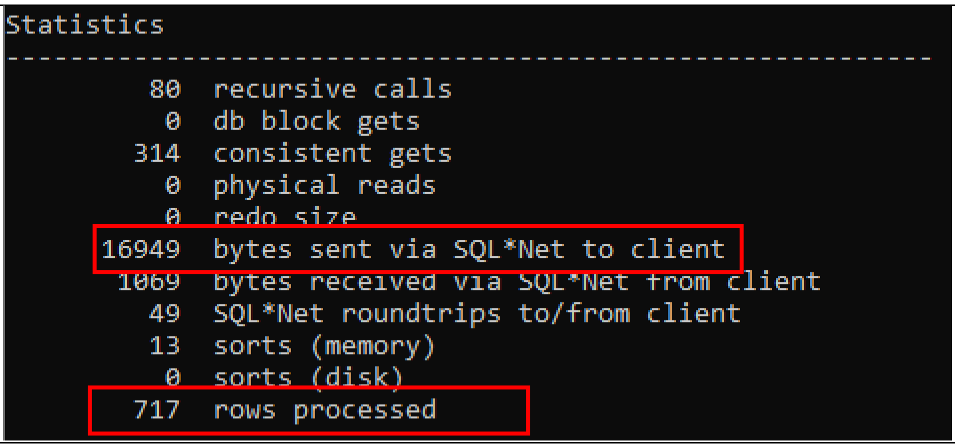
STEP 2

Time Elapsed:



```
Elapsed: 00:00:00.86
```

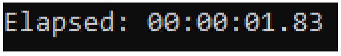
Statistical Output:



```
Statistics
----------------------------------------------------------
         80  recursive calls
          0  db block gets
        314  consistent gets
          0  physical reads
          0  redo size
      16949  bytes sent via SQL*Net to client
       1069  bytes received via SQL*Net from client
         49  SQL*Net roundtrips to/from client
         13  sorts (memory)
          0  sorts (disk)
        717  rows processed
```

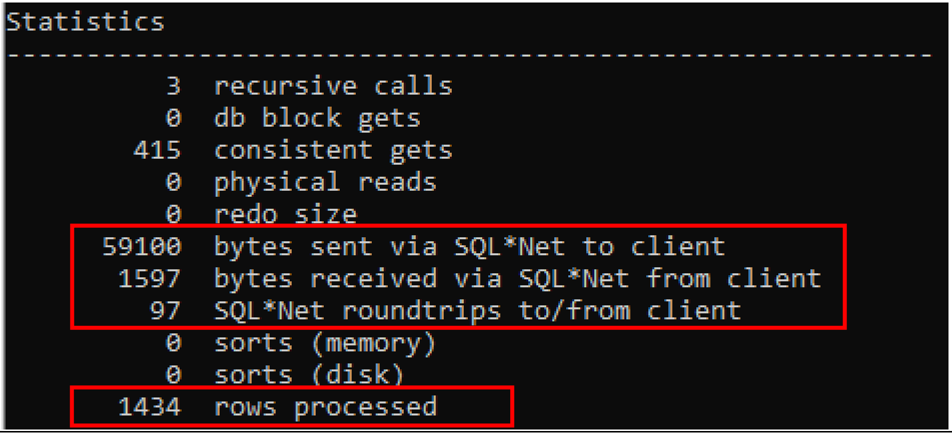
STEP 3

Time Elapsed:



```
Elapsed: 00:00:01.83
```

Statistical Output:



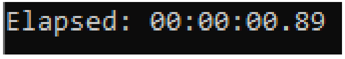
```
Statistics
----------------------------------------------------------
          3  recursive calls
          0  db block gets
        415  consistent gets
          0  physical reads
          0  redo size
      59100  bytes sent via SQL*Net to client
       1597  bytes received via SQL*Net from client
         97  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
       1434  rows processed
```

- **NOTE**: Key Factors are outlined in Red

Inner Joins

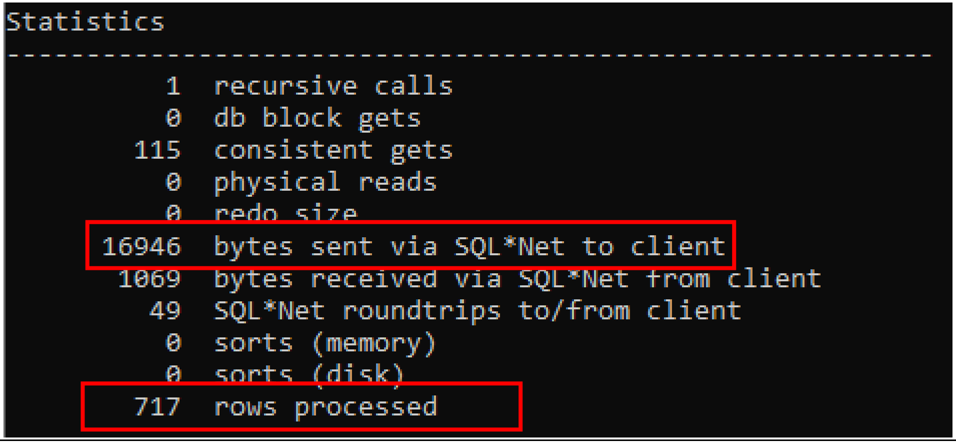
STEP 1

Time Elapsed:



```
Elapsed: 00:00:00.89
```

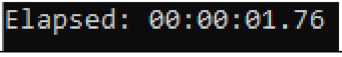
Statistical Output:



```
Statistics
----------------------------------------------------------
          1  recursive calls
          0  db block gets
        115  consistent gets
          0  physical reads
          0  redo size
      16946  bytes sent via SQL*Net to client
       1069  bytes received via SQL*Net from client
         49  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
        717  rows processed
```

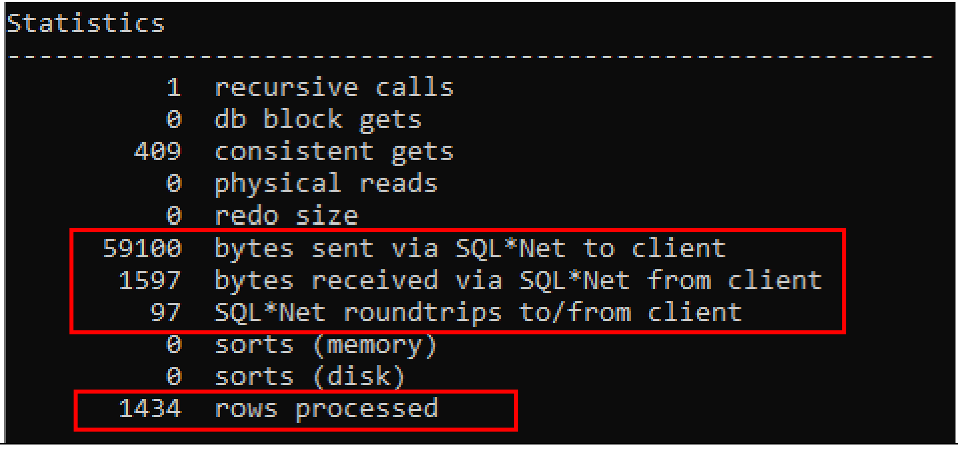
STEP 2

Time Elapsed:



```
Elapsed: 00:00:01.76
```

Statistical Output:

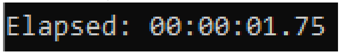


```
Statistics
----------------------------------------------------------
          1  recursive calls
          0  db block gets
        409  consistent gets
          0  physical reads
          0  redo size
      59100  bytes sent via SQL*Net to client
       1597  bytes received via SQL*Net from client
         97  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
       1434  rows processed
```

- **NOTE**: Key Factors are outlined in Red

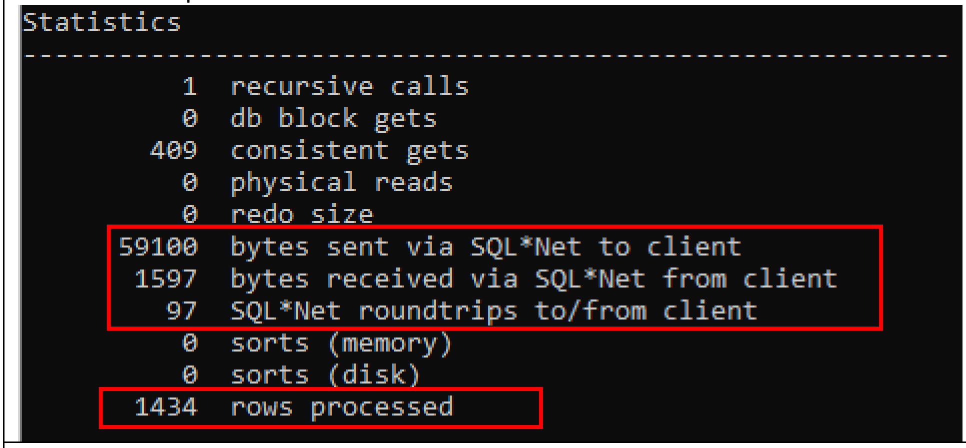
Global Join

Time Elapsed:



```
Elapsed: 00:00:01.75
```

Statistical Output:



```
Statistics
----------------------------------------------------------
          1  recursive calls
          0  db block gets
        409  consistent gets
          0  physical reads
          0  redo size
      59100  bytes sent via SQL*Net to client
       1597  bytes received via SQL*Net from client
         97  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
       1434  rows processed
```

NOTE: Key Factors are outlined in Red

In accordance with the task sheet, the method to calculate the transmission cost is as follows for both the semi-join and the inner-join plan. It is important to note that the most significant value for this calculation is the “bytes sent via SQL*NET to client” stated in the statistics. Hence, the calculations would be:

- Transmission Cost of the Semi Join Plan = 455 587 (step 1) + 16949 (step 2) = **472 536 bytes**
- Transmission Cost of the Inner Join Plan = **16 946 bytes** (step 1 only)

Therefore, the inner join plan is much better suited with regards to the global query, as the transmission cost is lower by 455 590 bytes in comparison to the semi join plan:

Transmission Cost of the Semi Join Plan − Transmission Cost of the Inner Join Plan
= 472 536 − 16 946 = **455 590 bytes**

Similarly, the task sheet also states that the final step of the join execution plan, regardless of the method of join – either semi-join or inner-join in this case – needs to be equivalent. Hence, Step 3 of the semi-join plan, Step 2 of the inner-join plan and the global join should all match up – specifically in the statistics' categories outlined in red.

The following table compares and summaries the important statistics of the join execution plans for these three methods:

	Semi-Join Plan – Step 3	**Inner-Join Plan – Step 2**	**Global Join Plan**
Bytes sent via SQL*NET to client	59 100	59 100	59 100
Bytes received via SQL*NET to client	1597	1597	1597
SQL*NET roundtrips to/from client	97	97	97
Rows Processed	1434	1434	1434
Elapsed Time	00:00:01.83	00:00:01.76	00:00:01.75

It is evident that majority of the important statistics in the three join execution plans for the final step are the same and reflect the equivalence of the respective queries. The most discerning factor here would be that the elapsed time is faster for the inner join plan as compared with the semi join plan by about 7 milliseconds. This is primarily due to the use of the intermediate that the semi join plan has (Step 1, 2), while the inner join's Step 1 statistics are already very similar to semi join's Step 2 statistics.

Hence, it is evident that the inner join is the better suited plan, given the lower data transmission cost, the lower time elapsed and the use of less intermediates.

References

Introduction to PL/SQL Anonymous Block. (2020, April 11). Retrieved March 22, 2023, from https://www.oracletutorial.com/plsql-tutorial/plsql-anonymous-block/