

# InnerJoin

1) Display the Cartesian product of the party and candidate table

The screenshot shows a SQL Developer window with a query in the Query Builder:

```
SELECT * FROM party_alalusi, candidate_alalusi;
```

The Query Result tab displays 24 rows, which is the Cartesian product of the two tables. The columns are: PARTYID, PARTYDESC, LNAME, FNAME, ADDRESS, SALARY, DOB, and PARTYID\_1.

PARTYID	PARTYDESC	LNAME	FNAME	ADDRESS	SALARY	DOB	PARTYID_1
1	Democrat	jennet	abraham	Berkeley, CA.	20000	01-FEB-60	1
2	Democrat	Green	abraham	Oakland, CA.	30000	01-FEB-64	1
3	Democrat	gren	cheryl	Berkeley, CA.	(null)	01-FEB-68	2
4	Democrat	greenr	albert	Salt Lake City, UT	40000	01-FEB-70	2
5	Democrat	gran	anne	Salt Lake City, UT	50000	01-FEB-61	3
6	Democrat	mama	mia	pepper City, UT	60000	01-FEB-68	(null)
7	Republican	jennet	abraham	Berkeley, CA.	20000	01-FEB-60	1
8	Republican	Green	abraham	Oakland, CA.	30000	01-FEB-64	1
9	Republican	gren	cheryl	Berkeley, CA.	(null)	01-FEB-68	2
10	Republican	greenr	albert	Salt Lake City, UT	40000	01-FEB-70	2
11	Republican	gran	anne	Salt Lake City, UT	50000	01-FEB-61	3
12	Republican	mama	mia	pepper City, UT	60000	01-FEB-68	(null)
13	Independent	jennet	abraham	Berkeley, CA.	20000	01-FEB-60	1
14	Independent	Green	abraham	Oakland, CA.	30000	01-FEB-64	1
15	Independent	gren	cheryl	Berkeley, CA.	(null)	01-FEB-68	2
16	Independent	greenr	albert	Salt Lake City, UT	40000	01-FEB-70	2
17	Independent	gran	anne	Salt Lake City, UT	50000	01-FEB-61	3
18	Independent	mama	mia	pepper City, UT	60000	01-FEB-68	(null)
19	4 (null)	jennet	abraham	Berkeley, CA.	20000	01-FEB-60	1
20	4 (null)	Green	abraham	Oakland, CA.	30000	01-FEB-64	1
21	4 (null)	gren	cheryl	Berkeley, CA.	(null)	01-FEB-68	2
22	4 (null)	greenr	albert	Salt Lake City, UT	40000	01-FEB-70	2
23	4 (null)	gran	anne	Salt Lake City, UT	50000	01-FEB-61	3
24	4 (null)	mama	mia	pepper City, UT	60000	01-FEB-68	(null)

2) Display the lastname and the party description of each individual

The screenshot shows a SQL Developer window with a query in the Query Builder:

```
SELECT lname, partydesc  
FROM candidate_alalusi, party_alalusi  
WHERE party_alalusi.partyid=candidate_alalusi.partyid;
```

The Query Result tab displays 5 rows, which are the results of the inner join. The columns are: LNAME and PARTYDESC.

LNAME	PARTYDESC
1 jennet	Democrat
2 Green	Democrat
3 gren	Republican
4 greenr	Republican
5 gran	Independent

3) Display the last name and the party description of each individual. If there is not a party description, then display no description. (Use the NVL function)

The screenshot shows the SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

```
01  
02 SELECT lname, NVL(partydesc, 'NoDESC')  
03 FROM candidate_alalusi, party_alalusi  
04 WHERE party_alalusi.partyid=candidate_alalusi.partyid OR party_alalusi.partyid=NULL;  
05  
06
```

The 'Query Result' tab shows the results of the query, with 5 rows fetched in 0.043 seconds. The columns are 'LNAME' and 'NVL(PARTYDESC,NODESC)'. The results are:

LNAME	NVL(PARTYDESC,NODESC)
1 jennet	Democrat
2 Green	Democrat
3 gren	Republican
4 greeenr	Republican
5 gran	Independent

4) Display the number of people in each party (display party\_description)

The screenshot shows the SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

```
08 FROM party_alalusi, candidate_alalusi  
09 WHERE party_alalusi.partyid=candidate_alalusi.partyid  
10 GROUP BY partydesc  
11 HAVING COUNT(*)>0;  
12  
13
```

The 'Query Result' tab shows the results of the query, with 3 rows fetched in 0.045 seconds. The columns are 'PARTYDESC' and 'PEOPLE'. The results are:

PARTYDESC	PEOPLE
1 Republican	2
2 Democrat	2
3 Independent	1

5) Display the number of people in each party for only those parties whose average salary is greater than 50,000. (Identify the party name)

The screenshot shows a SQL query editor with the following query:

```
93  
94  
95 SELECT partydesc, AVG(NVL(salary,0)),COUNT(*) People  
96 FROM party_alalusi, candidate_alalusi  
97 WHERE salary>50000 AND party_alalusi.partyid=candidate_alalusi.partyid  
98 GROUP BY partydesc  
99 HAVING AVG(salary)>50000 AND COUNT(*)>0;  
100  
101  
102
```

Below the query editor, the 'Query Result' tab is active, showing the following columns:

PARTYDESC	AVG(NVL(...	PEOPLE
-----------	-------------	--------

The status bar indicates 'All Rows Fetched: 0 in 0.035 seconds'.