Group By

Note: Display the party code, not the party description in each of the following

1) Display the number of people in each party. Order by Party. (Make sure to display the party_code)

```
SELECT partyid, COUNT(*)
FROM chave_candidate
GROUP BY partyid
ORDER BY partyid;
```

	PARTYID	COUNT(*)
1	1	2
2	2	2
3	3	1
4	(null)	1

2) Display the number of people in each party whose first name starts with d or r. (Make sure to display the party_code)

```
SELECT partyid, COUNT(partyid)
FROM chave_candidate
WHERE LOWER(fname) LIKE 'd%' OR LOWER(fname) LIKE 'r%'
GROUP BY partyid;
```

	PARTYID	COUNT(*)

3) Display the average salary for each party (Make sure to display the party_code)

```
SELECT partyid, AVG(salary)
FROM chave_candidate
GROUP BY partyid;
```

	PARTYID	AVG(SALARY)
1	1	25000
2	2	40000
3	(null)	60000
4	3	50000

4) Display the number of people in each party where the number of people does not exceed 2

```
SELECT partyid, COUNT(partyid)
FROM chave_candidate
GROUP BY partyid
HAVING NOT(COUNT(*)>2);
```

	PARTYID	COUNT(PARTYID)
1	1	2
2	2	2
3	(null)	0
4	3	1

5) Display the average salary for each party where the average does not exceed 50000

```
SELECT partyid, AVG(salary)
FROM chave_candidate
GROUP BY partyid
HAVING NOT(AVG(salary)>50000);
```

	PARTYID	AVG(SALARY)
1	1	25000
2	2	40000
3	3	50000

6) Create a new table called candidate2 that contains the number of people in each party. Should contain the partycode and the number of people (CAUTION, you have to use an alias for this to work)

```
DROP TABLE chave_candidate2;
CREATE TABLE chave_candidate2 AS SELECT partyid party_code, COUNT(partyid) num_of_people
FROM chave_candidate
GROUP BY partyid;
SELECT * FROM chave_candidate2;
```

	PARTY_CODE	NUM_OF_PEOPLE
1	1	2
2	2	2
3	(null)	0
4	3	1

Eliminating nulls

```
DROP TABLE chave_candidate2;
CREATE TABLE chave_candidate2 AS SELECT partyid party_code, COUNT(partyid) num_of_people
FROM chave_candidate
WHERE partyid IS NOT NULL
GROUP BY partyid;
SELECT * FROM chave_candidate2;
```

	PARTY_CODE	NUM_OF_PEOPLE
1	1	2
2	2	2
3	3	1